



Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

Proyecto de diseño de sistema electrónico para compartir ubicación entre dos satélites cubo

Realizado por Guillermo Nicolás **BUSTAMANTE**

Supervisado por Javier **GAGO BARRIO**

10 de octubre de 2018

Resumen

En este proyecto encontraremos el desarrollo del sistema electrónico de un nano satélite con respecto a la obtención de datos de temperatura, presión y ubicación, lo cuales, se transfieren, mediante comunicación inalámbrica y utilizando el protocolo I2C.

El proyecto se llevará a cabo en varias etapas. De un lado, la programación de un micro-controlador en la plataforma de código abierto Arduino, el cual gobernará sobre los sensores (presión y temperatura) y los módulos de ubicación GPS y comunicación Bluetooth.

Por otro lado, el diseño y fabricación de la Printed Circuit-Board (PCB), respetando las dimensiones del nano satélite ya disponible en el laboratorio de electrónica. Este modelo de nanosatélite es conocido como CubeSat, tiene forma de cubo de 10 cm de arista y pesa menos de 1,33kg.

El ensamblaje de la PCB con el micro-controlador y los componentes Arduino formarán el sistema de comunicación que será capaz de captar, procesar y enviar datos mediante una serie de sensores y módulos a un ordenador o dispositivo con tecnología Bluetooth, ambos con display para poder mostrar los datos recibidos.

Resum

En aquest projecte hi trobarem el desenvolupament del sistema electrònic d'un nano satèl·lit pel que fa a l'obtenció de dades de temperatura, pressió i ubicació, el quals, es transfereixen, mitjançant comunicació sense fils i utilitzant el protocol I2C.

El projecte es durà a terme en diverses etapes. D'una banda, la programació d'un micro-controlador a la plataforma de codi obert Arduino, el qual governarà sobre els sensors (pressió i temperatura) i els mòduls d'ubicació GPS i comunicació Bluetooth.

D'altra banda, el disseny i fabricació de la Printed Circuit-Board (PCB), respectant les dimensions del nano satèl·lit ja disponible al laboratori d'electrònica. Aquest model de nanosatèl·lit és conegut com CubeSat, té forma de cub de 10 cm d'aresta i pesa menys de 1,33kg.

L'acoblament de la PCB amb el micro-controlador i els components Arduino formaran el sistema de comunicació que serà capaç de captar, processar i enviar dades mitjançant una sèrie de sensors i mòduls a un ordinador o dispositiu amb tecnologia Bluetooth, tots dos amb display per poder mostrar les dades rebudes.

Abstract

In this project we will find the development of the electronic system of a nanosatellite with respect to the obtaining of data of temperature, pressure and location, which, are transferred, through wireless communication and using the I2C protocol.

The project will be carried out in several stages. On the one hand, the programming of a microcontroller on the open-source Arduino platform, which will govern the sensors (pressure and temperature) and the GPS location and Bluetooth communication modules.

On the other hand, the design and manufacture of the Printed Circuit Board (PCB), respecting the dimensions of the nanosatellite already available in the electronics laboratory. This model of nanosatellite is known as CubeSat, has cube shape of 10 cm edge and weighs less than 1.33 kg.

The assembly of the PCB with the microcontroller and the Arduino components will form the communication system that will be able to capture, process and send data through a series of sensors and modules to a computer or device with Bluetooth technology, both with a display to be able show the received data.

“Para Marquitos, Carlitos, Lidia, Jaz, Neli, Wal, Geri y Yonny”

Índice

Capítulo I.	Introducción	1
1.	Objeto del trabajo	1
2.	Alcance	1
3.	Especificaciones básicas	2
4.	Justificación	2
5.	Antecedentes y estado del arte.....	3
5.1.	Tecnología para pequeños vehículos espaciales (Small spacecraft technology) ..	3
5.2.	Estado del arte.....	4
5.3.	Proyecto CubeSat	5
Capítulo II.	Plan de trabajo	8
1.	Arduino y protoboard.....	8
2.	Diseño PCB (Printed Circuit-Board)	9
3.	Implementación en CubeSat	9
Capítulo III.	Componentes empleados.....	10
1.	Microcontrolador Arduino Nano	10
2.	Bluetooth HC-06	11
3.	Módulo GPS Neo-6MV2	12
4.	Sensor Barométrico BMP180	12
5.	Otros.....	13
Capítulo IV.	Implementación en Arduino.....	14
1.	Configuración Bluetooth HC-06.....	15
1.1.	Datos de programación y configuración	15
1.2.	Escritura y compilación del código	16
1.3.	Conexiones físicas.....	17
1.4.	Comprobación respuesta deseada	19
2.	Configuración GPS Neo-6MV2.....	21
2.1.	Datos de programación y configuración	21
2.2.	Escritura y compilación del código	21
2.3.	Conexiones físicas.....	23
2.4.	Comprobación respuesta deseada	24

3.	Configuración sensor barométrico BMP180	27
3.1.	Datos de programación y configuración	27
3.2.	Escritura y compilación del código	28
3.3.	Conexiones físicas.....	29
3.4.	Comprobación respuesta deseada.....	30
4.	Diseño menú opciones de usuario	32
Capítulo V.	Diseño PCB.....	33
1.	Introducción	33
2.	Diagrama de proceso.....	34
3.	Diseño.....	35
3.1.	Símbolos y Librerías.....	35
3.2.	Diagrama electrónico	36
3.3.	Huellas	37
3.4.	Netlist	38
3.5.	Diseño placa	38
3.6.	Producción PCB.....	40
3.7.	Vista 3D.....	41
Capítulo VI.	Implementación en el CubeSat	42
1.	Placa PCB recibida	42
2.	Modificaciones	44
3.	Implementación en CubeSat	45
Capítulo VII.	Conclusiones.....	48
Presupuesto.....		49
Bibliografía.....		50
Anexos		52
1.	Normas y recomendaciones en el layout de la PCB	52
2.	Código sistema de comunicaciones Arduino.....	54
3.	BOSCH BMP 180 Digital Pressure Sensor – DATASHEET [15]	56
4.	Módulo bluetooth HC06 – DATASHEET [16]	85
5.	GPS NEO 6MV2 – u-blox – DATASHEET [17].....	102
6.	Arduino nano – DATASHEET [18].....	127

Índice de figuras

Imagen 1. Dimensiones CubeSat.....	2
Imagen 2. Clasificación de los pequeños satélites según su masa. NASA (imagen KICKSAT [4])	3
Imagen 3. Diagrama de barras de nanosatélites lanzados.....	6
Imagen 4. Deforestación de más de una hectárea resaltada en líneas blancas, detectada en Pará Brasil de Julio a agosto. [9].....	6
Imagen 5. Nanosatélites creados por SAS	7
Imagen 6. Arduino nano V3	10
Imagen 7. Module Bluetooth HC-06.....	11
Imagen 8. GPS Modulo Ublox Neo 6m V2.....	12
Imagen 9. Sensor barométrico BMP180	13
Imagen 10.- Código configuración módulo Bluetooth HC-06	16
Imagen 11.- Configuración Bluetooth en IDE Arduino y respuesta en monitor serie.....	17
Imagen 12.- Imagen prototipo de conexión entre módulo Bluetooth, Arduino Nano y protoboard	18
Imagen 13.- Verificación de conexión entre el módulo bluetooth y el PC.	19
Imagen 14.- Módulo bluetooth configurado, conectado y con comunicación con PC.	20
Imagen 15.- Trama de datos NMEA.....	21
Imagen 16.- Inclusión de librerías NMEA en Arduino	22
Imagen 17.- Conversión de millas por horas a metros por segundo.....	22
Imagen 18.- Conversión de pies a metros.....	22
Imagen 19.- Utilización de librerías NMEA para la obtención de los datos del módulo GPS Neo-6mV2.....	23
Imagen 20.- Prototipo realizado en Fritzing de conexionado entre Bluetooth, GPS y Arduino.	24
Imagen 21.- Lectura de datos GPS a través del módulo Bluetooth HC-06.....	25
Imagen 22.- Imagen localización GPS proporcionada por el módulo GPS Neo 6mV2.....	25
Imagen 23.- Módulo GPS Neo-6mV2 y Bluetooth Hc-06 conectados al Arduino Nano y transmitiendo datos.	26
Imagen 24.- Comunicación I2C (source: https://teslabem.com/learn/fundamentos-del-protocolo-i2c-aprende/)	27
Imagen 25.- Trama de datos protocolo I2C (source: https://bit.ly/2zZeVUs)	27
Imagen 26.- Sistema de conexionado de todos los componentes y el Arduino nano mediante la protoboard y diseñado en Fritzing.....	30
Imagen 27.- Lectura de datos en Bluetooth Serial Terminal del sensor barométrico BMP180.	30
Imagen 28.- Temperatura Barcelona según la información de 'Google'	31
Imagen 29.- Prototipo de sistema de comunicación definitivo basado en la protoboard, Arduino y los módulos empleados	31
Imagen 30.- Menú usuario.....	32
Imagen 31.- Menú Usuario Serial Terminal	32
Imagen 32.- Esquemático KiCAD (source: http://kicad-pcb.org/)	33

Imagen 33.- PCB Layout KiCAD (source: http://kicad-pcb.org/).....	33
Imagen 34.- 3D Viewer KiCAD (source: http://kicad-pcb.org/)	34
Imagen 35.- Arduino Nano Librería y footprint.....	35
Imagen 36.- Ejemplo de módulo Bluetooth para recreación 3D	36
Imagen 37.- Esquemático Arduino Nano	37
Imagen 38.- Distribución sockets 'qbkit' de 'Open Cosmos'.....	37
Imagen 39.- Netlist. Asociación de huellas con componentes	38
Imagen 40.- PCB Version 1.....	39
Imagen 41.- Diseño final PCB.....	39
Imagen 42.- Vista previa a la fabricación de la PCB en Aisler.net.....	40
Imagen 43.- Vista posterior PCB en 3D Viewer KiCAD	41
Imagen 44.- Imagen 23.- Vista inferior PCB en 3D Viewer KiCAD.....	41
Imagen 45.- Placa PCB recibida pendiente de soldar componentes y de modificaciones ...	42
Imagen 46.- Soldado de pines de la PCB.....	43
Imagen 47.- Resistencias SMD	43
Imagen 48.- Puente para corregir error de conexionado	44
Imagen 49.- Diseño Final PCB.	44
Imagen 50.- Test final lectura de datos con PCB integrada en CubeSat.....	45
Imagen 51.- Comunicación final de lectura de datos.....	46
Imagen 52.- Diseño Final de PCB implementada en CubeSat.....	47

Capítulo I. INTRODUCCIÓN

1. Objeto del trabajo

El objetivo principal del trabajo es diseñar e implementar un sistema de bajo coste que obtenga información de la ubicación de un CubeSat, donde irá implementado, la temperatura y la presión atmosférica, y la transmita a otro similar. Este sistema será capaz de captar, procesar y enviar datos, a un ordenador o dispositivo con tecnología Bluetooth, a través de una serie de sensores gobernados por un micro-controlador Arduino.

El sistema en cuestión, constará de un micro-controlador Arduino, un sensor de temperatura y presión, un módulo de localización GPS y un módulo comunicación inalámbrica. Los cuales irán ensamblados en una PCB diseñada y fabricada a priori.

El sistema deberá cumplir una serie de requisitos:

- Bajo coste, con la intención de minimizar el coste económico del proyecto.
- De código y hardware abierto para que sea de libre acceso a ediciones.
- Adaptarse a las dimensiones del nanosatélite diseñado en el laboratorio de electrónica para continuar el trabajo creado en proyectos anteriores y poder expandirse en futuros proyectos.
- Ser un sistema autónomo capaz de procesar, recibir y enviar datos cuando algún dispositivo se conecte a este.

2. Alcance

El alcance de este proyecto es el desarrollo y análisis de un sistema electrónico basado en Arduino capaz de recibir y transmitir datos a través de comunicación inalámbrica cumpliendo con las especificaciones técnicas del proyecto CubeSat.

Por consiguiente, podríamos dividir el alcance del proyecto en tres diferentes fases:

- *Programación del micro-controlador Arduino* para una correcta comunicación y transmisión de datos captados a través de sensores y módulos.
- *Diseño e implementación de la Printed Circuit-Board (PCB)* cumpliendo con las especificaciones técnicas.
- *Implementación del sistema en el CubeSat.*

3. Especificaciones básicas

Para asegurar y mantener la eficacia del sistema diseñado se han implementado las siguientes especificaciones básicas:

- La Printed Circuit-Board (PCB) deberá adaptarse al molde del nanosatélite disponible en el laboratorio de electrónica. El cual responde a los estándares del CubeSat. [1]

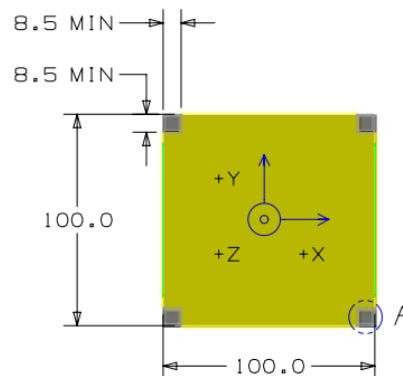


Imagen 1. Dimensiones CubeSat

- La conexión de los sensores y los módulos se realizará a través de las pistas de la Printed Circuit-Board (PCB) para reducir el cableado. Esto conlleva a un mejor aprovechamiento del espacio.
- Las Printed Circuit-Board se han diseñado siguiendo las especificaciones de la norma IPC2221 que define los requisitos generales para el diseño de PCBs.
- El micro-controlador deberá poder operar con un voltaje de 5V o menos para reducir el consumo de energía.
- El diseño final tendrá en cuenta 2 aspectos importantes:
 - o Deberá disponer de un total de cuatro entradas y salidas digitales y 2 salidas y entradas analógicas.
 - o Deberá dejar libre algunas entradas y salidas tanto digitales como analógicas dando la posibilidad a una expansión del proyecto.
- El sistema deberá ser compatible con Windows y/o Android por su uso más expandido y de fácil acceso.
- El diseño del sistema ensamblado deberá facilitar las modificaciones y continuidad del desarrollo del CubeSat.

4. Justificación

Este diseño e implementación sobre el desarrollo del sistema electrónico del CubeSat permitirá avanzar una de las fases del proyecto en el que participan diversos estudiantes de diferentes ámbitos de la ingeniería de la UPC.

Mediante el uso correcto de una programación, ensamblaje, diseño e implementación de componentes electrónicos económicos y de código abierto se facilita la posibilidad a futuras modificaciones de forma sencilla y económica.

En la parte personal, ser partícipe de un proyecto de estas dimensiones me da una visión de los futuros proyectos en los que podría participar o realizar como ingeniero. Además, me otorga la satisfacción de colaborar en un trabajo en el que han estado involucradas muchas ramas de la ingeniería (espacial, industrial, electrónica, entre otras...)

5. Antecedentes y estado del arte

5.1. TECNOLOGÍA PARA PEQUEÑOS VEHÍCULOS ESPACIALES (SMALL SPACECRAFT TECHNOLOGY)

En los últimos años el interés en los pequeños satélites ha incrementado debido a su bajo coste y tiempo de desarrollo. Y las nuevas tecnologías como la nanotecnología y los microsistemas electromagnéticos y la microelectrónica han hecho posible el diseño de estos pequeños satélites. [2]

Hoy en día hay varios tipos de satélites que entran dentro de la categoría de pequeños vehículos espaciales (small spacecraft) clasificados según su masa, tal y como muestra la figura de abajo. Existen los mini satélites con una masa de 100 a 500 kg; los microsátélites, de 10 a 100 kg; los nanosatélites, de 1 a 10 kg; y los picos satélites con una masa menor de 1 kg. [3]

Como ejemplo de límite de masa superior se encuentra el mini satélite FASTSAT (Fast, Affordable, Science and Technology Satellite) lanzado en 2010 con una masa ligeramente por debajo de 180kg. Y en el extremo más bajo, el KICKSAT con una masa por debajo de los 10 gramos.

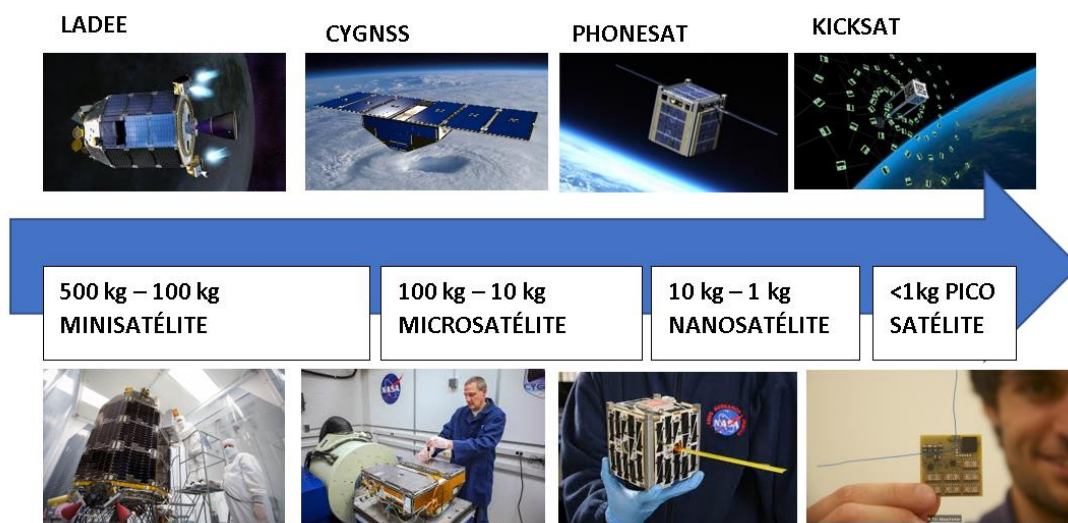


Imagen 2. Clasificación de los pequeños satélites según su masa. NASA (imagen KICKSAT [4])

5.2. ESTADO DEL ARTE

Para que sea correcta la funcionalidad de los pequeños satélites es necesario que haya un equilibrio entre el tamaño y la tecnología empleada, por lo que la miniaturización e integración son aspectos importantes.

En los siguientes dos apartados se explica brevemente la miniaturización e integración.

Miniaturización tecnológica - Microsistemas

Un ejemplo de miniaturización tecnológica son los microsistemas. Son sistemas inteligentes que abarcan funciones de sensores, de procesamiento o de actuación con la finalidad de incorporarlos en aparatos electrónicos.

En Europa se denomina *Tecnología de Microsistemas* - MST, *Micro máquinas* en Japón y la más conocida *Sistemas micro electro mecánicos* - MEMS en América.

Como su nombre lo indica sus componentes son de dimensiones en la escala de micras (μ).

En 1960, científicos demostraron que, fusionando varios micro transistores en un solo chip, se podía armar circuitos electrónicos a esa escala, mejorando la funcionalidad, rendimiento y fiabilidad, mientras se reducía su coste y volumen. Llegando así a demostrar una respuesta superable a la de sus homólogos en macro escala. [5]

Integración - Circuito integrado

Un circuito integrado es un conjunto de elementos miniaturizados que componen un sistema más complejo para realizar una función electrónica específica.

Se fabrica con un material semiconductor sobre el cual se incluyen los circuitos electrónicos a través de la fotolitografía. Se protegen estos circuitos con un encapsulado con conectores metálicos que permiten la conexión entre la pastilla semiconductora y el circuito.

A finales de 1950, el ingeniero Jack St. Clair Kilby desarrolló el primer prototipo de circuito integrado para la compañía Texas Instruments. Los equipos electrónicos solían incorporar tubos al vacío, también llamados válvulas electrónicas o termoiónicas. Estos tubos se usaban para conmutar, modificar o amplificar una señal electrónica dirigiendo el movimiento de los electrones mediante ciertos gases o en un espacio con una presión muy reducida. Por tanto, el desarrollo de Kilby redujo el espacio de trabajo de estos tubos, ubicando estos componentes activos y pasivos en una misma superficie de metal de pequeñas dimensiones.

En la actualidad, existen varios tipos de circuitos integrados, un ejemplo son los microprocesadores que se encuentran tanto en ordenadores como en móviles. [6]

5.3. PROYECTO CUBESAT

Una vez introducidos los avances que permiten la fabricación de dispositivos en una escala reducida, se explica el proyecto del CubeSat.

El proyecto CubeSat empezó en 1999 como un proyecto de colaboración entre el Prof. Jordi Puig - Suari de la Universidad Politécnica del Estado de California (Cal Poly) y el Prof. Bob Twiggs del departamento de Desarrollo de Sistemas Espaciales de la Universidad de Stanford (SSDL).

El propósito de este proyecto es crear unas normas estándares para reducir el coste y tiempo de desarrollo de los pequeños satélites y así, mejorar e incrementar la accesibilidad al espacio manteniendo frecuentes lanzamientos. Actualmente, el Proyecto CubeSat se colabora en más de 100 universidades, institutos y otras instituciones.

OBJETIVO

El programa CubeSat tiene como principal misión permitir el acceso de pequeñas cargas al espacio, bajo la responsabilidad del Cal Poly de garantizar la seguridad del CubeSat y proteger el vehículo de lanzamiento (Launch Vehicle, LV).

Para más información acerca del proyecto CubeSat consultar las referencias [1]

HISTORIA

Desde el primer lanzamiento de un CubeSat al espacio en junio del 2003 sobre un Russian Eurockot, hasta el 2013, la mayoría de estos lanzamientos eran hechos por proyectos de las universidades y actividades de investigación. A partir de entonces, la mayoría de los lanzamientos son con propósitos no académicos, sino comerciales o como proyectos de principiantes. [7]

Mientras aumenta su popularidad, se crean varias compañías que proveen de materiales y kits para su fabricación como paneles solares, estructuras, baterías. Un ejemplo de estas compañías son OpenCosmos, Tyvak, ISIS, Pumpkin Inc, etc.

En el siguiente gráfico se muestra los CubeSat que están en órbita (verde), los lanzamientos fallidos (rojo), los futuros lanzamientos (azul) y las predicciones desde enero de 2018 para ese año (gris).

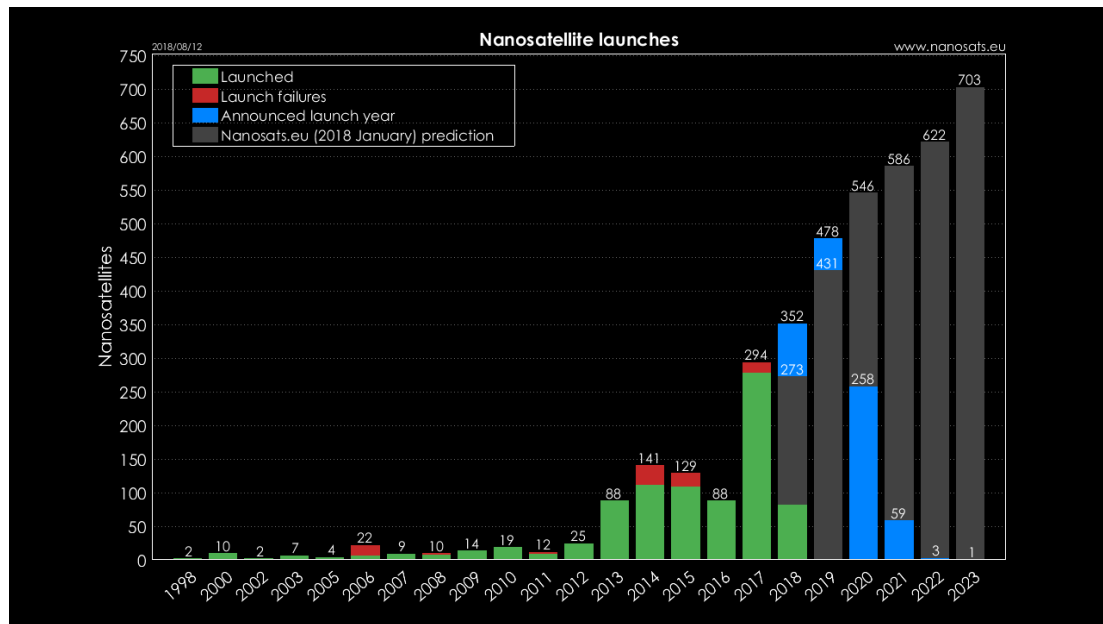


Imagen 3. Diagrama de barras de nanosatélites lanzados

APLICACIONES

OBSERVACIÓN DE LA TIERRA (EARTH OBSERVATION, EO)

Dentro de la aplicación EO se puede encontrar servicios relacionados con la meteorología, agricultura de precisión o varios tipos de monitorización. Un ejemplo de desarrollo de esta aplicación es realizada por la compañía *Planet*; *Deforestation Alert* (Alerta de deforestación).

Esta aplicación ayuda a sus clientes a controlar la deforestación mediante alertas que informan cuando la deforestación ha alcanzado el máximo predefinido. [8]



Imagen 4. Deforestación de más de una hectárea resaltada en líneas blancas, detectada en Pará Brasil de Julio a agosto. [9]

SISTEMAS DE GEOLOCALIZACIÓN

Un ejemplo de geolocalización podría ser un seguimiento del tránsito aéreo y barcos, ya que normalmente están para trabajar alrededor del aeropuerto, puertos y zonas pobladas. Pero es escasa en áreas oceánicas o polos. Al utilizar estos CubeSat aumenta la cobertura de rastrear de manera fiable los aviones y barcos que se encuentran sobre zonas remotas, evitando así perder el seguimiento y control de éstos. [8]

Un ejemplo de aplicación lo ofrece la compañía *Karten Space*, Control Marítimo que ofrece:

- Seguridad y vigilancia
- Búsqueda y rescate de embarcaciones
- Investigación de incidentes
- Control y lucha contra la piratería
- Traqueo general de barcos
- Mejora de gestión de puertos
- Asistencia en la navegación [10]

COMUNICACIONES

Respecto a las comunicaciones mediante CubeSat, se incluye a la compañía *Sky and Space Global* (SAS). Su proyecto consiste en el despliegue de constelaciones de nanosatélites (aprox. 200), colocados en órbita cuidadosamente seleccionadas, para lograr cubrir total cobertura alrededor del ecuador de la Tierra, creando así una red de comunicación por voz, datos y mensajería instantánea.

Servicios del SAS:

- Proveedores de comunicación satelital de banda ancha
- Seguimiento a tiempo real para aviones y compañías navieras
- Servicio complementario para redes móviles en regiones de América Latina, Asia y África con poca cobertura
- Entre otros [11]

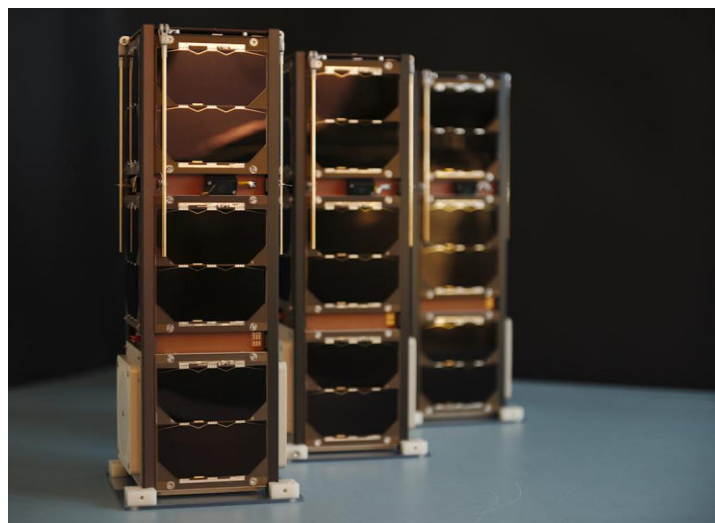
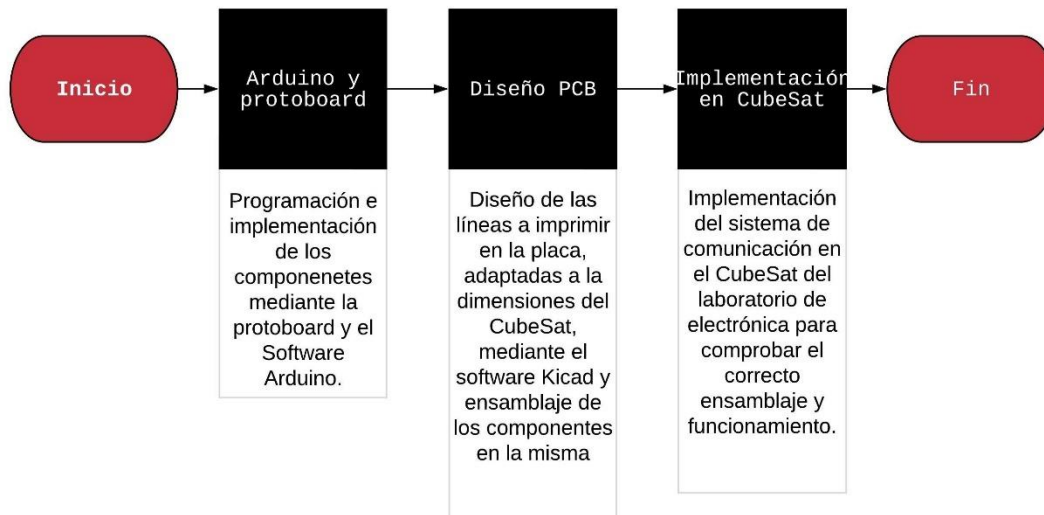


Imagen 5. Nanosatélites creados por SAS

Capítulo II. PLAN DE TRABAJO

Una vez planteados los objetivos del proyecto y analizados los diferentes puntos en el estado del arte, detallaremos el plan de trabajo a seguir:



1. Arduino y protoboard

Este capítulo pretende abarcar todas las fases que contemplan la configuración y programación de los componentes del proyecto con el objetivo de obtener las lecturas de presión, temperatura y ubicación mediante Bluetooth. Para conseguir esto será necesario una protoboard, para la realización de las conexiones físicas entre el micro-controlador Arduino y los componentes, y la utilización del software Arduino, donde programaremos y configuraremos los componentes.

Pensando en un trabajo posterior, en un primer momento se había optado por emplear un Arduino Mega, debido a la cantidad de entradas y salidas que maneja respecto al Arduino Nano. Esto permitiría en un futuro ampliar la cantidad de componentes empleados. Pero, por las dimensiones del Arduino Mega no es posible su implementación en un CubeSat, midiendo este 101 mm, mayor que la dimensión permitida para un CubeSat. Por tanto, se ha optado por la implementación del Arduino Nano que, sí que cumple con estos requisitos de dimensiones y, a parte, para el estudio del presente proyecto consta con las suficientes entradas y salidas, permitiendo, incluso, ampliar en 2 componentes más.

2. Diseño PCB (Printed Circuit-Board)

El objetivo de esta parte es el diseño de una placa de circuito impreso que cumpla con las normas generales de fabricación de una PCB y que el dimensionado sea acorde a la cavidad disponible en el CubeSat del laboratorio. Esto, permite la substitución del cableado por una placa compacta y funcional.

La realización del diseño se hace por medio del software KiCAD. Explicado con más detalle en el propio capítulo.

3. Implementación en CubeSat

Obtenido el correcto funcionamiento y conexionado del sistema formado por la PCB, el Arduino y los componentes, se procederá a la instalación del mismo en el CubeSat.

Antes de proceder a la implementación de la PCB en el CubeSat, se hará una breve explicación de las modificaciones realizadas en la PCB post producción.

Capítulo III. COMPONENTES EMPLEADOS

1. Microcontrolador Arduino Nano

El microcontrolador es el principal elemento en nuestro proyecto, ya que es el responsable de la ejecución de las órdenes grabadas en su memoria y el que gobierna el resto de componentes del sistema.

De entre la multitud de placas de diferentes fabricantes (Raspberry Pi, BeagleBone, Nanode...), que son comparables con los microcontroladores Arduino, nos hemos declinado por esta por los siguientes motivos que consideramos ventajosos para este proyecto:

- El lenguaje de programación (IDE) es fácil de usar y aprender
- Las tarjetas son económicas. En nuestro caso hablamos de una tarjeta con un monto no superior a los 30€
- Software open-source. Permitiendo el desarrollo a programadores de nuevas librerías para uso público.
- Está extendido lo que permite obtener sus componentes en multitud de tiendas de electrónica.
- Tienen una gran comunidad. Esto, enriquece la documentación.
- Cumple con todas las necesidades del proyecto.

En la ilustración 1 podemos observar el modelo de la placa de Arduino, en concreto, el Arduino Nano, el cual utilizaremos para este proyecto:

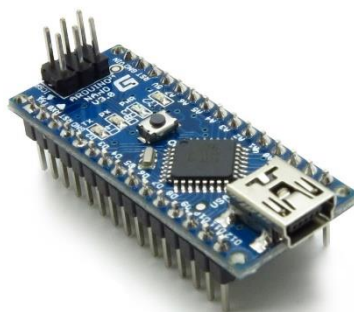


Imagen 6. Arduino nano V3

A continuación, se pueden observar las principales características técnicas:

Tabla I.- Características técnica Arduino Nano

Características	Descripción
Procesador	ATmega 328P
Velocidad de Reloj	16MHz
Tensión de alimentación	7V a 12V
Tensión de trabajo	5V
Número de entradas analógicas	6
Número de entradas digitales	14

2. Bluetooth HC-06

Dado que la gran mayoría de dispositivos bluetooth disponibles en el mercado tienen características similares, a la hora de la elección, nos hemos basado en uno de los más populares, el Bluetooth HC-06.

Dentro de la familia del HC-06 también encontramos el bluetooth HC-05. La diferencia entre ambos es que el primero solo puede conectarse como esclavo y el segundo, puede hacerlo como Maestro o Esclavo. Para el presente proyecto, nuestro maestro será un PC o dispositivo, por tanto, no necesitamos las funciones de maestro en el bluetooth.

Volviendo al Bluetooth HC-06, este presenta una dirección MAC única de 48 bits, un nombre para poder identificarlo fácilmente y un pin para acceder a él. El nombre y pin son fácilmente editables a través de comando AT de configuración, en capítulos posteriores se explicará en detalle.

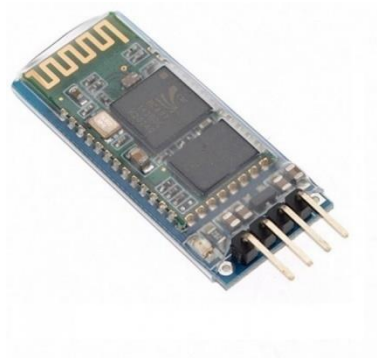


Imagen 7. Module Bluetooth HC-06

Dentro de sus características principales podemos destacar:

- Compatible con el protocolo Bluetooth V2.0
- Voltaje de alimentación: 3.3 - 6VDC
- Voltaje de operación: 3.3VDC.
- Baud rate ajustable: 1200, 2400, 4800, 9600, 38400, 57600, 115200
- Tamaño: 4.4 cm x 1.6 cm x 0.7 cm

3. Módulo GPS Neo-6MV2

Para la obtención de los datos de ubicación se ha optado por la utilización del módulo GPS NEO-6MV2.

Los dispositivos NEO-6VM2 son una familia de receptores fabricados por U-Blox, que pueden ser conectados con facilidad a un procesador como Arduino. Este receptor se ha elegido para nuestro proyecto debido a su diseño de pequeñas dimensiones, poco coste y poco consumo. La intensidad de medición es de unos 37mA en modo continuo.

La precisión en posición es de unos 2.5m, en velocidad de 0.1 m/s y en orientación de 0.5º, valores más que aceptables para nuestro proyecto.



Imagen 8. GPS Modulo Ublox Neo 6m V2

Para más información se adjunta el datasheet en el Anexo 5.

4. Sensor Barométrico BMP180

El sensor barométrico BMP180 es un módulo de la marca Bosch de alta precisión con un rango de medida de entre los 300 y los 1100hPa con un margen de error de 0.03hPa. Está diseñado para la lectura de la presión atmosférica, temperatura y, a raíz de estas, estimar la altura. Su diseño está optimizado para aplicaciones en las que el consumo eléctrico se requiere bajo, como es el caso del CubeSat. Además, sus dimensiones son pequeñas (3.60x3.80x0.93 mm) y es de bajo coste.

Incorpora una interfaz I²C para permitir la comunicación con Arduino. Está compuesto por un sensor BMP180, un regulador de Voltaje (5V a 3.3V), resistencia pull-up y capacitadores by-pass. En la siguiente imagen podemos observar su forma y diseño:

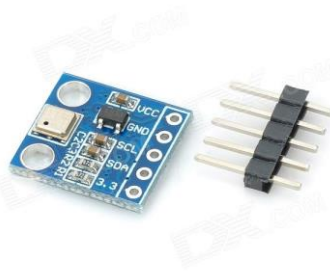


Imagen 9. Sensor barométrico BMP180

Sus características principales son las siguientes:

- Interfaz digital con dos cables (I²C)
- Amplio rango de medición atmosférica
- Medición de temperatura incluida
- Ultra-bajo consumo de energía
- Bajo ruido
- Ultraplano y pequeñas dimensiones

Para más información se adjunta el datasheet en el Anexo 3.

5. Otros

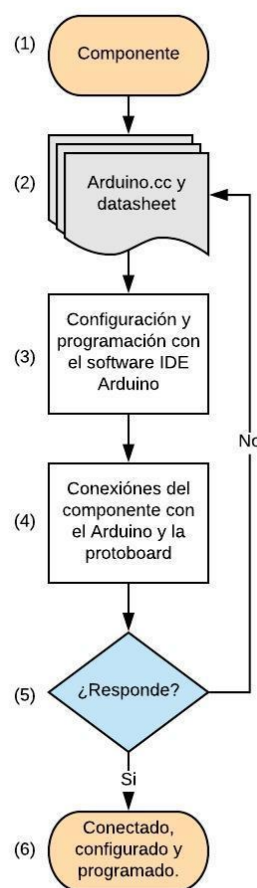
Otras herramientas utilizadas en el proyecto y que se pueden incluir en la parte de componentes necesarios para la realización del mismo son las siguientes:

1. *Protoboard y cables.* Para la realización de la conexión física entre el Arduino Nano y los módulos.
2. *Arduino.* es una plataforma electrónica de código abierto basada en hardware y software de electrónica.
3. *KiCAD.* Software de diseño *open-source* formado por varios programas que trabajan en conjunto para ayudar al usuario en el diseño de la PCB.
4. *Fritzing.* Herramienta 'open source' de múltiples funcionalidades. La funcionalidad elegida para este proyecto ha sido la del diseño de los esquemas de comunicación amigables de la protoboard con Arduino y los módulos.
5. *Bluetooth Serial Terminal.* Para la realización de los pasos de configuración de dispositivos será necesaria la instalación de una de las múltiples aplicaciones de comunicación serial que sea capaz de enviar y recibir comandos. Bluetooth Serial Terminal se encarga de esto.
6. *Alimentación externa.* Aunque no es necesario, a la hora de conseguir las lecturas del GPS es mejor tener el mismo en zonas abiertas. Por ello, una alimentación externa permitirá mantener el prototipo alimentado mientras se hacen las pruebas y sin necesidad de tenerlo conectado a enchufes.

Capítulo IV. IMPLEMENTACIÓN EN ARDUINO

En este capítulo procederemos a la explicación detallada de la programación y conexionado de cada componente sobre el Arduino Nano.

Para facilitar la comprensión del lector hemos diseñado el siguiente 'flowchart' donde se refleja el proceso general que seguiremos para cada componente.



- (1) Componente a configurar.
- (2) Extracción de los datos de configuración y programación del sitio web oficial www.Arduino.cc y del datasheet del fabricante.
- (3) Escritura y compilación del código en el software IDE Arduino.
- (4) Realización de las conexiones físicas entre el Arduino Nano y el componente.
- (5) Comprobación de que la respuesta es la deseada.
- (6) El componente queda configurado y programado.

1. Configuración Bluetooth HC-06

1.1. DATOS DE PROGRAMACIÓN Y CONFIGURACIÓN

Para programar este componente se han utilizado las siguientes funciones relevante que permiten la comunicación en serie entre el módulo Bluetooth HC-06 y el Arduino Nano.[12]

1. *Serial.begin(9600)*: Abre el puerto serie y establece la velocidad en bits por segundo (baudios) para la transmisión de datos en serie.
2. *Serial.available()*: Obtiene la cantidad de bytes (caracteres) disponibles para leer desde el puerto serie.
3. Librería *SoftwareSerial*. El Arduino Nano viene de fábrica con dos puertos que permiten la comunicación en serie, RX y TX. Debido a que la comunicación con el PC y el bluetooth son en serie, necesitamos ampliar el número de puertos. Razón por la cual, implementamos el uso de esta librería. En este caso, le indicamos que debe crear los nuevos puertos en el pin 2 (RX) y 3 (TX) de nuestro Arduino nano.
4. Se utilizarán otras funciones que no son propias de la comunicación en serie, éstas, se explicarán en el código.

Para la configuración, el módulo Bluetooth HC-06 cuenta con comandos AT propios, ejecutables a través del puerto serie, con los cuales se puede modificar la que viene por defecto.

En este caso, se modificará la configuración de velocidad de comunicación y la configuración del nombre por defecto. La configuración del pin la hemos aplicado para confirmar.

Tabla II.- Datos comandos AT módulo Bluetooth HC-06

Comunicación Bluetooth HC-06 - TFG Guillermo N Bustamante				
Comando para cambiar valores por defecto				
	Por defecto	Deseado	COMANDO	RESPUESTA
Nombre	HC-06	BlueCubeSat	AT+NAMEBlueCubeSat	OKsetname
PIN	1234	1234	AT+PIN1234	OKsetpin
Velocidad	38400	9600	AT+BAUD4	OK9600

En la tabla anterior, los parámetros quedarían de la siguiente forma:

1. Nombre: BlueCubeSat
2. Pin: 1234
3. Velocidad: 38400

En el documento adjunto del Bluetooth HC06 se pueden consultar más comandos de configuración del módulo.

1.2. ESCRITURA Y COMPILACIÓN DEL CÓDIGO

Con las funciones explicadas con anterioridad y los comandos a utilizar procedemos a escribir el código y compilarlo.

```

1 //Configuración módulo Bluetooth HC-06
2 //Guillermo Nicolás Bustamante
3 //TFG - Octubre 2018
4
5 #include <SoftwareSerial.h> // Incluimos la librería SoftwareSerial
6                             // que nos permite ampliar el número de puertos
7                             // para la comunicación en serie
8
9 SoftwareSerial BlueCubeSat(2,3); // Asignamos al puerto virtual el nombre 'BlueCubeSat'
10                                // y definimos los pines 2 y 3 como el RX y TX de este puerto virtual.
11
12 void setup()
13 {
14   BlueCubeSat.begin(9600); // Inicializamos el puerto virtual 'BlueCubeSat' que hemos creado
15   Serial.begin(9600);      // Inicializamos el puerto serie
16 }
17
18 void loop()
19 {
20   if(BlueCubeSat.available()) // Si llega un dato por el puerto virtual 'BlueCubeSat' se envía al monitor serial
21   {
22     Serial.write(BlueCubeSat.read());
23   }
24
25   if(Serial.available()) // Si llega un dato por el monitor serial se envía al puerto virtual 'BlueCubeSat'
26   {
27
28     BlueCubeSat.write(Serial.read());
29   }
30 }
31

```

Imagen 10.- Código configuración módulo Bluetooth HC-06

La figura 10 es el código utilizado para enviarle comandos AT desde el monitor serial por el USB o puerto serie físico del Arduino al módulo bluetooth por el puerto virtual BlueCubesat con la finalidad de configurarlo. La respuesta del módulo bluetooth se lee con el puerto virtual creado en el Arduino y se retransmite por el puerto físico Serial para visualizarlo en el ordenador a través del monitor serial. Para ello, primero modificamos el nombre del puerto serie virtual para identificarlo más fácilmente, con el comando siguiente: BlueCubesat (2,3); el 2 y 3 hacen referencia a los pines que utilizaremos para el puerto virtual. Acto seguido, configuramos las velocidades del puerto virtual y puerto físico a 9600 baudios e iniciamos ambos puertos.

Una vez iniciados los puertos, se programa un 'loop' para permitir la lectura de los datos que llegan por el puerto serie físico Serial (comandos AT escritos y enviados desde el monitor serial del IDE de arduino) para transmitirlos por el puerto serie virtual del bluetooth BlueCubesat hacia el módulo Bluetooth.

El módulo Bluetooth recibe los comandos y envía a través del puerto virtual (3, TX) una respuesta, la cual es leída a través del monitor serial para verificar que la configuración ha sido exitosa.

En las siguientes imágenes se muestra la respuesta del módulo confirmándonos la modificación.

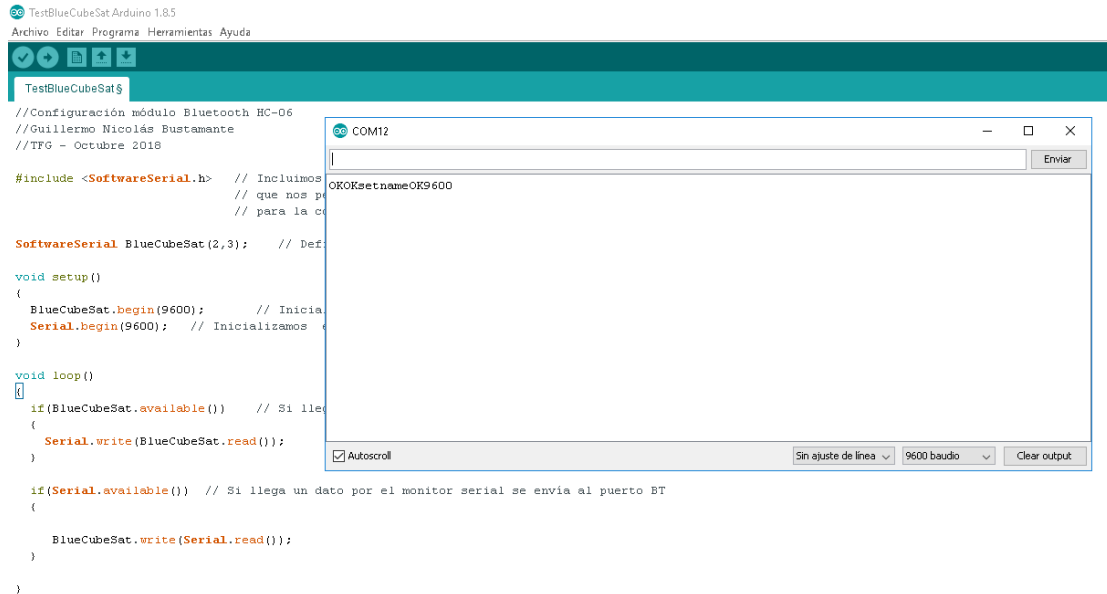


Imagen 11.- Configuración Bluetooth en IDE Arduino y respuesta en monitor serie.

Como observamos en la imagen anterior, las respuestas obtenidas coinciden con las esperadas. Detalladas en la Tabla II.

La modificación de la contraseña se ha realizado en el paso posterior a la captura de pantalla y la respuesta fue la esperada 'OKsetpin'.

Para la correcta lectura de datos es importante que, la velocidad de comunicación entre el puerto Bluetooth, el puerto serie del micro-contralor Arduino y el monitor serie, sean iguales. En el presente proyecto, la comunicación con los componentes se ha realizado a 9600 Baudios.

1.3. CONEXIONES FÍSICAS

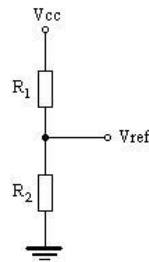
El módulo HC06 consta de cuatro pines, de los cuales dos son para la alimentación y dos para la comunicación en serie con nuestro Arduino Nano.

Descripción de pines:

- VCC: polo positivo de la alimentación (3.6 - 6 V)
- GND: Tierra
- TXD: Interfaz de comunicación serie, terminal de transmisión.
- RXD: Interfaz de comunicación serie, terminal de recepción (3.3 V)

Para la conexión del Pin RXD, se ha tenido en cuenta que su voltaje operativo es de 3.3V. La salida de los pines del Arduino Nano proporciona una tensión 5V. Por tanto, debemos reducir esta tensión para no quemar la entrada RXD.

Para ello, se ha creado un divisor de tensión. En el siguiente esquema se muestra cómo se compone un divisor de tensión:



Donde, V_{cc} hace referencia a la salida digital del pin 3 del Arduino Nano y V_{ref} a la entrada RXD del módulo Bluetooth.

La ecuación resultante del esquema es la siguiente:

$$V_{ref} = \frac{R_2 \times V_{cc}}{R_1 + R_2}$$

Si el V_{cc} es 5V, V_{ref} es 3.3V y nos inventamos una resistencia de $R_1 = 1\text{ kohm}$.

Despejando encontraríamos el valor que necesitamos de $R_2 = 2,2\text{ kohm}$

Una vez encontrados los valores de las resistencias, se puede proceder al montaje de las conexiones entre el Arduino nano y el módulo bluetooth. En la imagen siguiente realizada en Fritzing se observa cómo debería quedar el montaje.

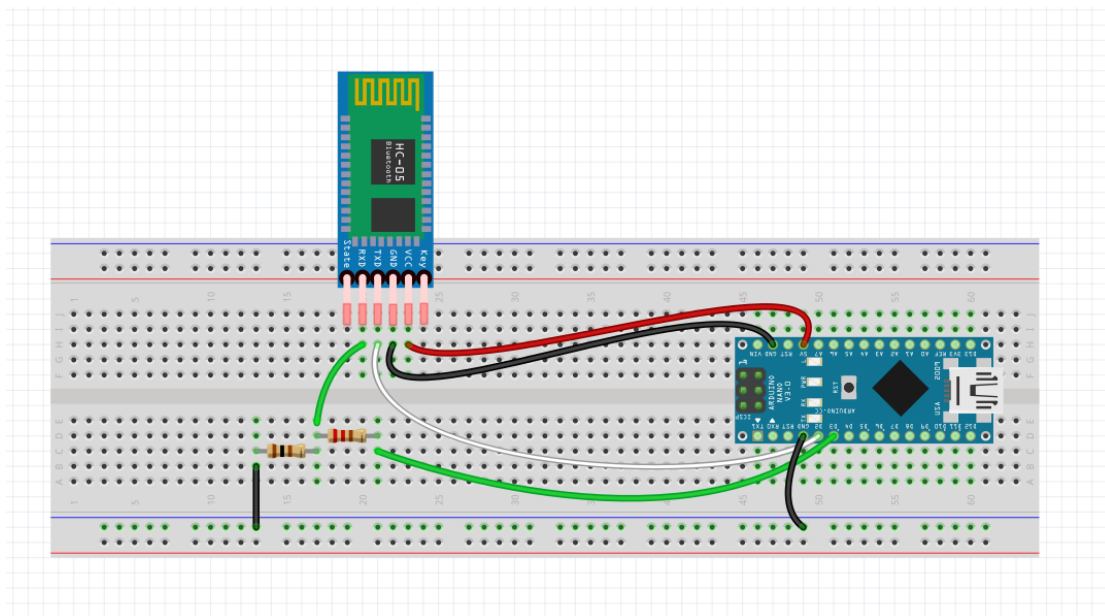


Imagen 12.- Imagen prototipo de conexión entre módulo Bluetooth, Arduino Nano y protoboard

Se ha utilizado un Bluetooth HC-05 debido a que el módulo HC-06 no figura en las librerías de Fritzing. El conexionado en ambos es el mismo. Por tanto, se ha proseguido con el conexionado explicado en pasos anteriores.

1.4. COMPROBACIÓN RESPUESTA DESEADA

El primer paso a seguir a la hora de utilizar un módulo Bluetooth, es el emparejamiento del mismo con el PC. Para ello debemos dirigirnos a la configuración de módulos Bluetooth del PC o teléfono y buscar el nombre del Bluetooth que se desea emparejar. Una vez cliques en emparejar, pedirá la contraseña, en nuestro caso '1234'.

Por otra parte, para la comprobación de la respuesta deseada necesitaremos descargar una aplicación que permita simular un terminal. En nuestro caso, se ha utilizado una aplicación para PC disponible en Windows, 'Bluetooth Serial Terminal'.

En la aplicación 'Bluetooth Serial Terminal' buscaremos el dispositivo que acabamos de configurar y emparejar, 'BlueCubeSat', y clicaremos en iniciar conexión. Si no está emparejado, no aparecerá en los módulos bluetooth disponibles.

Si la conexión es correcta, simplemente la aplicación te dejará la opción de desconectar, en caso contrario, aparecerá un mensaje de error. En la imagen siguiente, se observa como el Bluetooth se encuentra conectado al PC y disponible para el envío de datos.

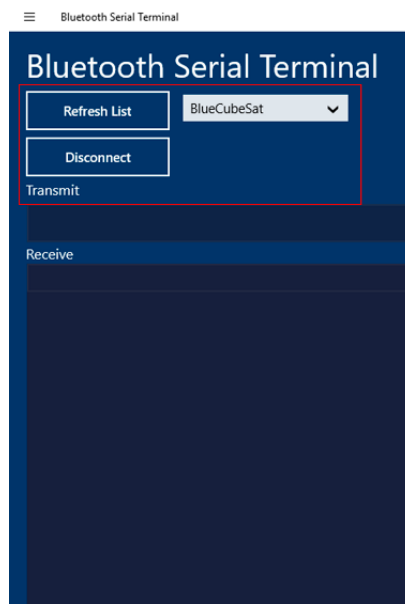


Imagen 13.- Verificación de conexión entre el módulo bluetooth y el PC.

Por otro lado, si el bluetooth está emparejado con el PC, el LED del módulo Bluetooth permanecerá encendido, sin parpadear. Esto significa, que ahora el módulo está disponible para enviar y recibir comandos. Cuando enviamos o recibimos datos los LEDs RX y TX del Arduino Nano parpadearan para informarnos.

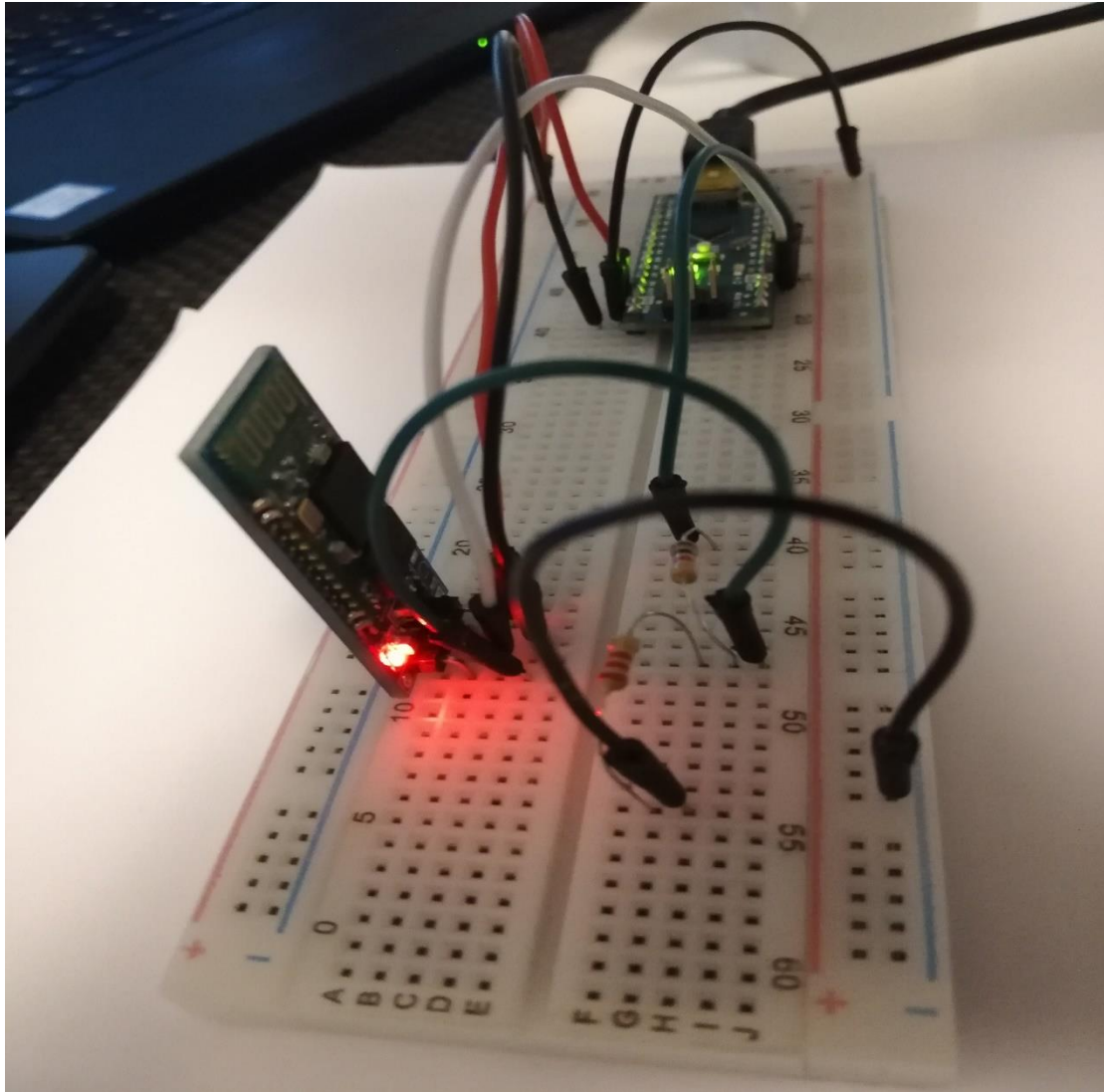


Imagen 14.- Módulo bluetooth configurado, conectado y con comunicación con PC.

2. Configuración GPS Neo-6MV2

2.1. DATOS DE PROGRAMACIÓN Y CONFIGURACIÓN

La comunicación entre el GPS Neo-6mV2 y el Arduino Nano es similar a la del módulo bluetooth, ambos se comunican a través del puerto serie. Por tanto, para llevar a cabo esta comunicación se necesitarán repetir los pasos del apartado de programación del bluetooth. Exceptuando incluir otra vez la librería SoftwareSerial, con una vez es suficiente. Se utilizarán los pines 8 (RX) y 9(TX) para la comunicación con el puerto serie del GPS.

Es importante destacar que, a diferencia del módulo Bluetooth, el módulo GPS no necesita configuración del módulo. Con la configuración de fábrica es suficiente para poder comunicarnos con él. En caso de ser necesaria una modificación por temas de incompatibilidades, de velocidad, por ejemplo, habría que mirar en el datasheet del fabricante. Adjunto en el Anexo 5.

Prosiguiendo, el GPS Neo-6mV2 recoge los datos a través de su antena integrada, y los envía en tramas separadas por comas y letras de identificación (Imagen 15), a este formato se le denomina *NMEA 0183* y es el formato utilizado a nivel mundial por todos los dispositivos de navegación satelital. Aunque podemos descifrar el contenido de los datos, el formato no es cómodo a la hora de trabajar.

```
$GPGGA,110617.00,41XX.XXXXX,N,00831.54761,W,1,05,2.68,129.0,M,50.1,M,,*42
```

Imagen 15.- Trama de datos NMEA

En Arduino se han desarrollado múltiples librerías para interpretar y transformar estos datos en un formato más amigable para el usuario. Un ejemplo de éstas, es la **NMEAGPS** que utilizaremos en este proyecto. Dicha librería consta de alto contenido en información y documentación, facilitando así su aplicación. En el siguiente apartado se hará uso de la librería y se podrán observar las ventajas que representa.

2.2. ESCRITURA Y COMPILACIÓN DEL CÓDIGO

La finalidad de este código es conseguir los datos de interés que deseamos del GPS mediante conexión inalámbrica y haciendo uso de las librerías NMEAGPS. En las siguientes figuras se muestran las partes más relevantes del código. El código completo se encuentra en el Anexo 2.

```
#include <NMEAGPS.h>           // Librería para parsear la información que
                                // recibimos del GPS

SoftwareSerial BlueCubeSat(2,3); // Definimos los pines 2 y 3 como RX y TX
SoftwareSerial GpsCubeSat(8,9);  // Definimos los pines 8 y 9 como RX y TX

NMEAGPS GpsObject;              // Creamos un objeto de la clase NMEAGPS y lo llamamos 'GpsObject'
gps_fix data;                   // Creamos una variable que contendrá todos los datos que obtengamos
                                // y la llamamos 'data'
```

Imagen 16.- Inclusión de librerías NMEA en Arduino

En el uso de la librería NMEA es necesaria la creación de objetos. Un objeto es un elemento que hereda todos los atributos de la clase a la cual pertenece y puede utilizar las funciones implementadas en dicha clase. Se ha creado el objeto 'GpsObject' y el objeto 'data'.

'GpsObject' pertenece a la clase NMEAGPS la cual es utilizada para las funciones de comunicación. Y, 'data', pertenece a la clase 'gps_fix' la cual es la encargada del procesamiento de datos.

Una vez implementadas las clases, es posible trabajar con ella como se muestra a continuación. Podemos acceder a la documentación para encontrar todos los tipos de datos que son posible obtener. Para este proyecto se ha visto necesaria la obtención de los siguientes valores:

1. Satélites: Número de satélites que orbitan la tierra localizados. Es necesario un mínimo de tres satélites para conocer la ubicación y un mínimo de cuatro para saber la altitud.
2. Latitud y longitud: Devolverá las coordenadas del CubeSat con lo cual podremos saber su ubicación.
3. Velocidad: Devuelve el valor de la velocidad en millas/h con la que el CubeSat se mueve.
4. Altitud: Altura a la cual se encuentra el CubeSat y devuelve el valor en pies.

Como los valores devueltos por la librería NMEAGPS son en sistema inglés (millas/h y pies), se han debido aplicar los siguiente valores de conversión con tal de obtener estos datos en el sistema internacional.

millas por hora ↗

1

 mph

↔

→

metros por segundo ↗

0.44704 m/s

Imagen 17.- Conversión de millas por horas a metros por segundo

pie ↗

1

 ft

↔

→

metro ↗

0.3048 m

≈ 304 mm 4/5 mm

Imagen 18.- Conversión de pies a metros

Después de aclarar todos los detalles de configuración, en la siguiente imagen se puede observar el código resultante.

```
if (GpsObject.available(GpsCubeSat)) { // Comprueba que el puerto Gps este operativo
    data = GpsObject.read();           // Lee todos los datos del gps y los registra en data
}

if(BlueCubeSat.available()) // Comprueba si ha llegado algún dato por el puerto BlueCubeSat
{
    char BluetoothData=BlueCubeSat.read(); // Guardamos el dato recibido en una variable
    if (BluetoothData=='1'){
        // Comprueba si el usuario ha presionado '1'
        BlueCubeSat.println("Satelites: (u)");
        BlueCubeSat.println( data.satellites );           // Número de Satélites que ha localizado e
        BlueCubeSat.println("Latitud:");
        BlueCubeSat.println( data.latitude(), 6);         // Latitud de ubicación de CubeSat
        BlueCubeSat.println("Longitud:");
        BlueCubeSat.println( data.longitude(), 6);        // Longitud de ubicación de Cubesat
        BlueCubeSat.println("Velocidad (m/s):");
        BlueCubeSat.println( data.speed_mph() *0.44704 ); // Velocidad del CubeSat. Multiplicamos po
        BlueCubeSat.println("Altura: (m)");
        BlueCubeSat.println( data.altitude() *0.3048);    // Altura del Cubesat.
        BlueCubeSat.println();
    }
}
```

Imagen 19.- Utilización de librerías NMEA para la obtención de los datos del módulo GPS Neo-6mV2

2.3. CONEXIONES FÍSICAS

El módulo GPS Neo-6mV2 consta de los mismos pines que el módulo Bluetooth Hc-06. En distinto orden y con la diferencia de que el RXD del GPS Neo-6mV2 acepta 5V. Por tanto, no es necesario un divisor de tensión.

Descripción de pines

- VCC: polo positivo de la alimentación (3.6 - 6 V)
- TXD: Interfaz de comunicación serie, terminal de transmisión.
- RXD: Interfaz de comunicación serie, terminal de recepción.
- GND: Tierra

En la imagen siguiente se muestra cómo debería quedar el conexionado de nuestro prototipo una vez añadido el módulo GPS.

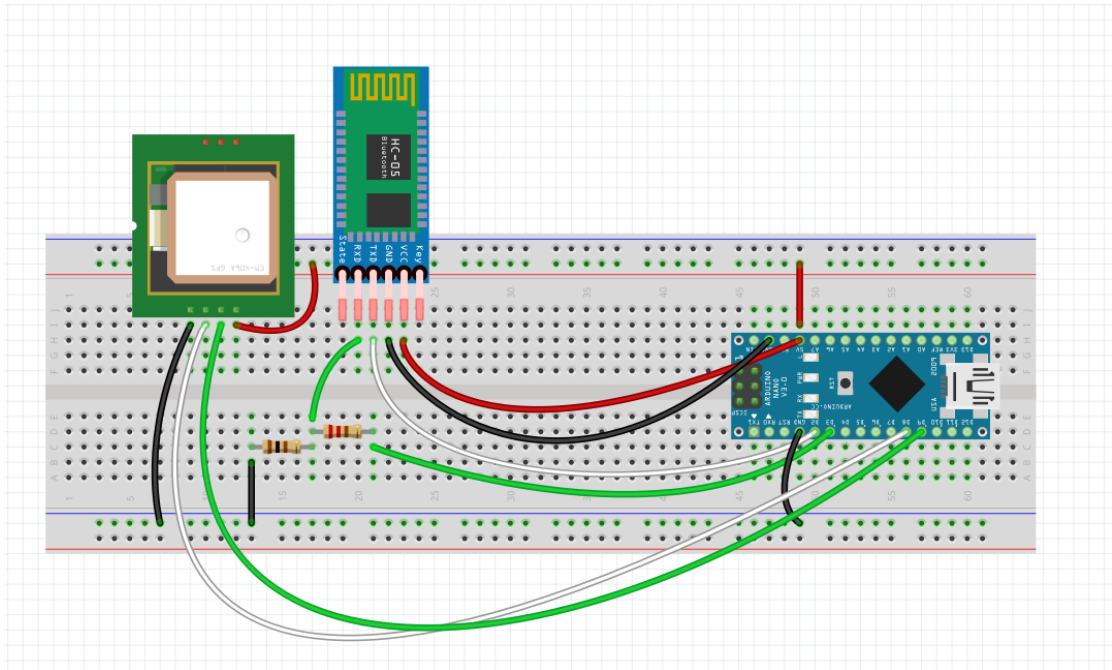


Imagen 20.- Prototipo realizado en Fritzing de conexionado entre Bluetooth, GPS y Arduino.

2.4. COMPROBACIÓN RESPUESTA DESEADA

A la hora de comprobar la respuesta deseada es importante ubicar el módulo GPS Neo-6mV2 en zonas donde pueda localizar otros satélites. La mejor ubicación es una zona abierta. A continuación, mostramos un ejemplo de la respuesta obtenida en una zona dentro de la casa frente a la que obtuvimos en una terraza.

La antena dentro de la casa no es capaz de captar los satélites, por eso, en caso no detectar como mínimo tres satélites, los datos recibidos serán cero.

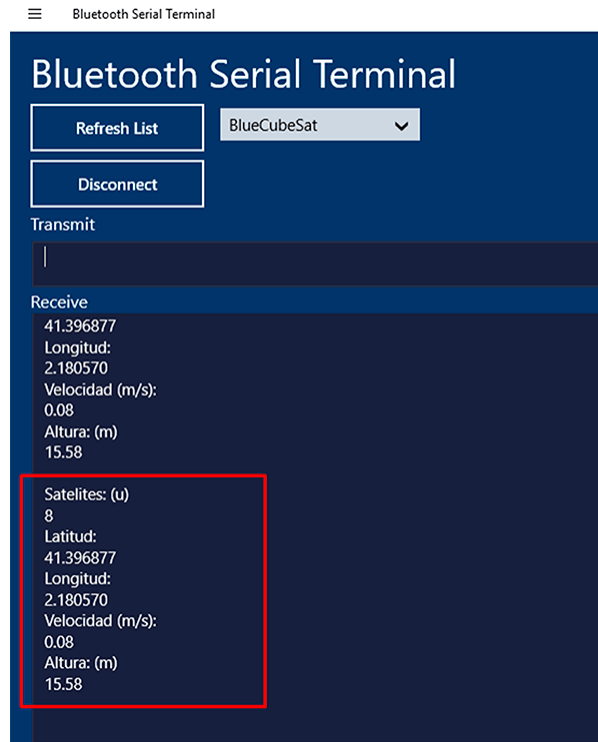


Imagen 21.- Lectura de datos GPS a través del módulo Bluetooth HC-06

Los datos recibidos en el Bluetooth serial terminal nos indican que el GPS ha localizado 8 satélites. Y, colocando los datos de latitud y longitud en <https://www.coordenadas-gps.com/> obtenemos la ubicación del módulo GPS.

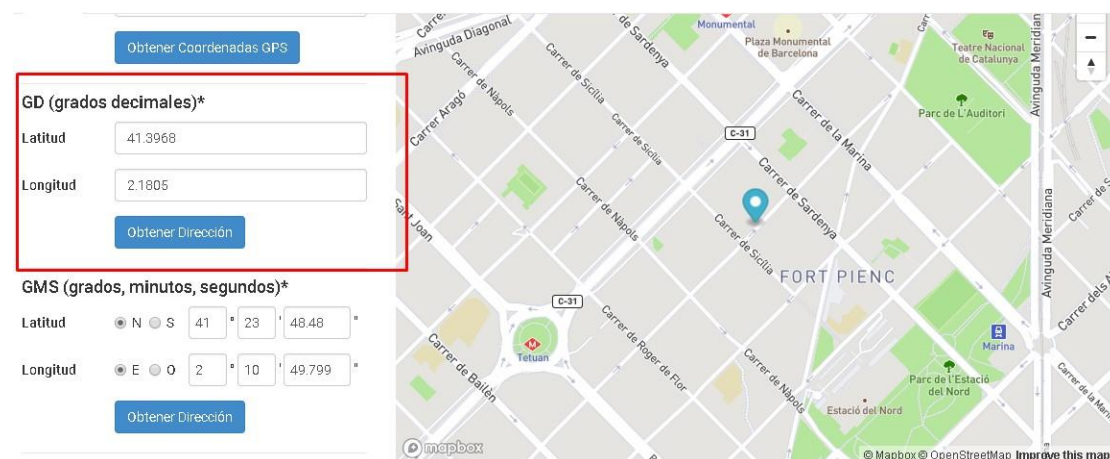


Imagen 22.- Imagen localización GPS proporcionada por el módulo GPS Neo 6mV2

Por otro lado, la velocidad es 0 m/s debido a que no estamos en movimiento y la altura es aproximada al piso donde nos encontramos.

El módulo GPS consta de una luz intermitente azul que nos informa que este está obteniendo información de los satélites. En el momento que la luz ya no aparezca, significa que no tiene señal o que está mal conectado.

Para finalizar este apartado se ha colocado una imagen de cómo han quedado los conexiones del Arduino, el bluetooth y el GPS Neo-6mV2 con la protoboard.

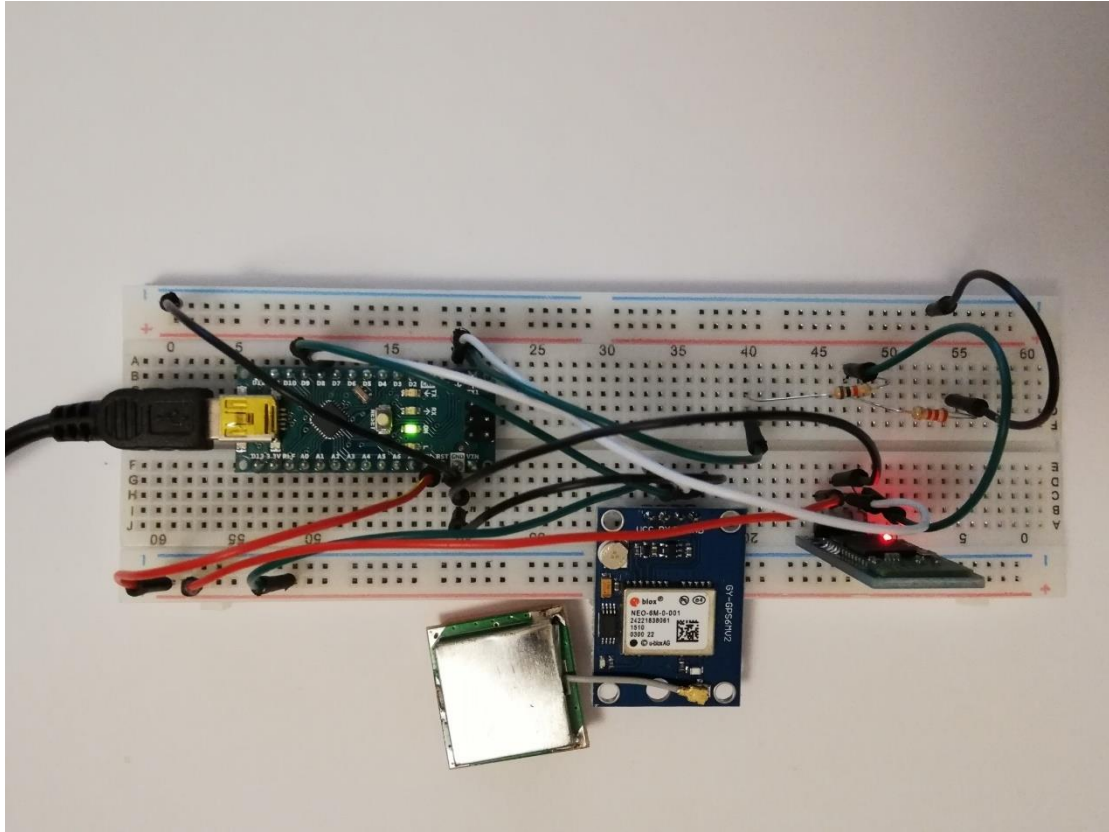


Imagen 23.- Módulo GPS Neo-6mV2 y Bluetooth Hc-06 conectados al Arduino Nano y transmitiendo datos.

3. Configuración sensor barométrico BMP180

3.1. DATOS DE PROGRAMACIÓN Y CONFIGURACIÓN

La comunicación del módulo BMP180 con el Arduino Nano es totalmente diferente a la de los anteriores módulos realizados. El sensor barométrico BMP180 es un esclavo y se comunica a través del bus I2C, el cual Arduino lleva incorporado.

Para la utilización del bus I2C debemos entender su protocolo. El protocolo I2C básicamente es la comunicación entre un maestro y uno o varios esclavos por medio de dos cables de comunicación. [13]

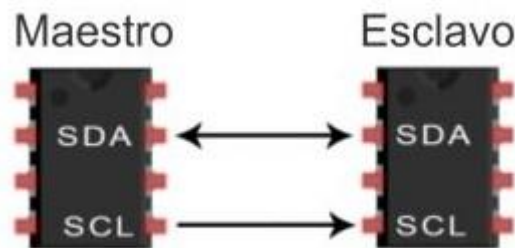


Imagen 24.- Comunicación I2C (source: <https://teslabem.com/learn/fundamentos-del-protocolo-i2c-aprende/>)

SDA – Serial Data. Vía para el envío de información entre el maestro y el esclavo

SCL – Serial Clock. Vía por donde viaja la señal de reloj.

Del datasheet del Arduino Nano se puede extraer los pines que hacen referencia al protocolo I2C. Concretamente, el pin analógico cuatro (A4) es el encargado de la transmisión de datos (SDA) y el pin analógico cinco (A5) es el encargado de la señal de reloj (SCL).

La información recibida viaja en trama de datos y cada trama lleva un mensaje compuesto por la dirección binaria del esclavo al que va dirigido el mensaje, como también, unos bits de inicio y paro, de lectura y escritura.

La forma de la trama se puede observar en la siguiente imagen:



Imagen 25.- Trama de datos protocolo I2C (source: <https://bit.ly/2zZeVUs>)

Para trabajar con las tramas del BMP180 Arduino incorpora unas librerías que facilitan su uso. Dentro las muchas librerías que existen, en este proyecto, se utilizará la librería **SFE_BMP180** desarrollada por SparkFun.

Las funciones básicas para el uso de la librería son las siguientes:

- `begin()`: Inicializa el sensor BMP180
- `startTemperature()`: Inicia la medición de temperatura. El retorno de esta función es el tiempo que tarda en medir la temperatura. El cual se deberá aplicar como delay para no provocar errores de lectura.
- `getTemperature(T)`: Obtiene el valor de la temperatura y lo almacena en la variable T.
- `startPressure()`: Inicia la medición de presión. Sucede lo mismo que con el `startTemperature()`, devuelve una variable de tiempo que se debe aplicar al delay.
- `getPressure(P,T)`: Obtiene el valor de la presión y lo almacena en la variable P. La temperatura en la medición de la presión es utilizada para compensar la influencia de esta en el cálculo de la presión. En el datasheet del Anexo 3 se puede encontrar información más detallada de los cálculos.

3.2. ESCRITURA Y COMPILACIÓN DEL CÓDIGO

El primer paso a la hora de empezar a trabajar con el módulo BMP180 en el software de programación Arduino es importar la librería. Como se ha explicado en el apartado anterior, la librería a importar es la **SFE_BMP180**.

```
TestBlueCubeSat_Gps$  
//Configuración módulo Bluetooth HC-06 + GPS Neo 6m  
//Guillermo Nicolás Bustamante  
//TFG - Octubre 2018  
  
#include <SoftwareSerial.h>    // Incluimos la librería SoftwareSerial  
                                // que nos permite ampliar el número de puer  
                                // para la comunicación en serie  
  
#include <NMEAGPS.h>          // Librería para parsear la información que  
                                // recibimos del GPS  
  
#include <SFE_BMP180.h>        // Incluimos librería de SparkFun  
SFE_BMP180 bmp180;            // Creamos un objeto de la clase SFE_BMP180
```

Una vez importada la librería es necesario, como en casos anteriores, la creación de un objeto con el cual poder aplicar las funciones y atributos de la clase `SFE_BMP180` (incluida en la librería con el mismo nombre)

Estos pasos previos nos permiten poder hacer uso de las funciones de lectura de presión y temperatura. En la imagen siguiente, mostramos los pasos, que también han sido explicados

con detalles en la sección anterior, de cómo conseguir una correcta lectura de la presión en mb y la temperatura en °C:

```

if (BluetoothData=='2'){                                     // Si presionamos 2 ...
    char status;
    double T,P;
    status = bmp180.startTemperature();                     //Inicio de lectura de temperatura
    if (status != 0){
        delay(status);                                     //Pausa para que finalice la lectura
        status = bmp180.getTemperature(T);                 //Obtener la temperatura
        if (status != 0){
            status = bmp180.startPressure(3);              //Inicio lectura de presión

            if (status != 0){
                delay(status);                             //Pausa para que finalice la lectura
                status = bmp180.getPressure(P,T);           //Obtenemos la presión

                if (status != 0){
                    Serial.print("Temperatura: ");
                    Serial.print(T,2);
                    Serial.print(" *C ");
                    Serial.print(" ");
                    Serial.print("Presion: ");
                    Serial.print(P,2);
                    Serial.println(" mb");
                }
            }
        }
    }
}
}
}
}

```

3.3. CONEXIONES FÍSICAS

El módulo BMP180 consta de cuatro pines los cuales dos son de alimentación y los otros dos son de comunicación I2C.

Descripción de pines:

- VCC: polo positivo de la alimentación (5 V)
- GND: Tierra
- SCL: Conexión de la señal de reloj al pin A5.
- SDA: Conexión de la señal de datos al pin A4.

En la siguiente imagen se muestra cómo debería quedar el prototipo definitivo de la protoboard con todos los componentes conectados:

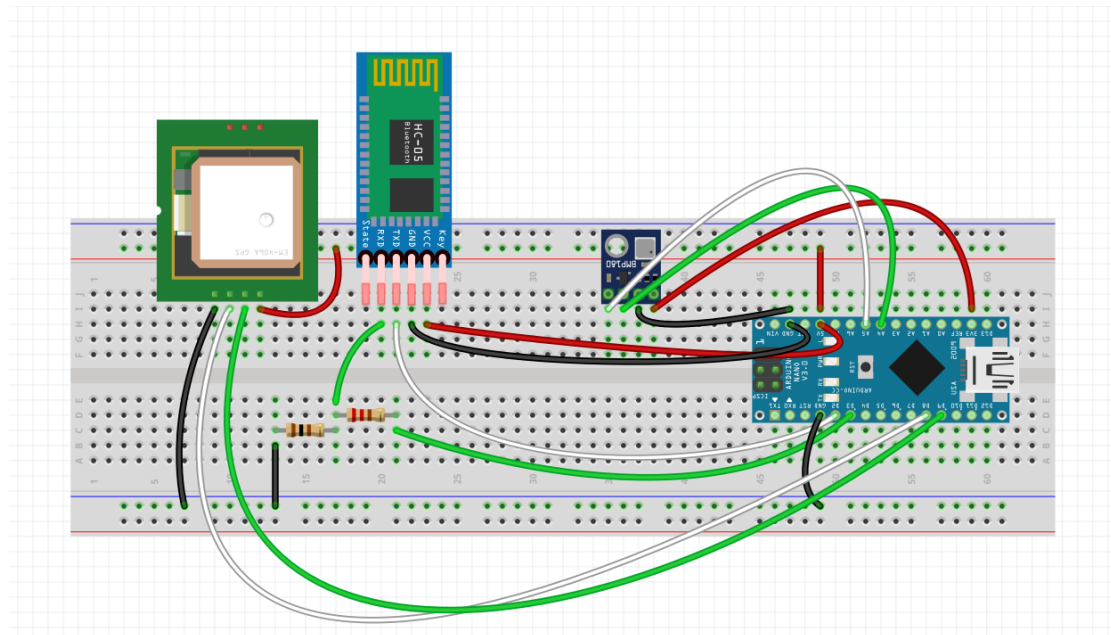


Imagen 26.- Sistema de conexionado de todos los componentes y el Arduino nano mediante la protoboard y diseñado en Fritzing.

3.4. COMPROBACIÓN RESPUESTA DESEADA

En este apartado es relativamente sencillo comprobar si el funcionamiento es correcto y la medición de la temperatura tiene precisión. En la siguiente imagen se puede observar como son los datos obtenidos a través de la aplicación 'Bluetooth Serial Terminal'.

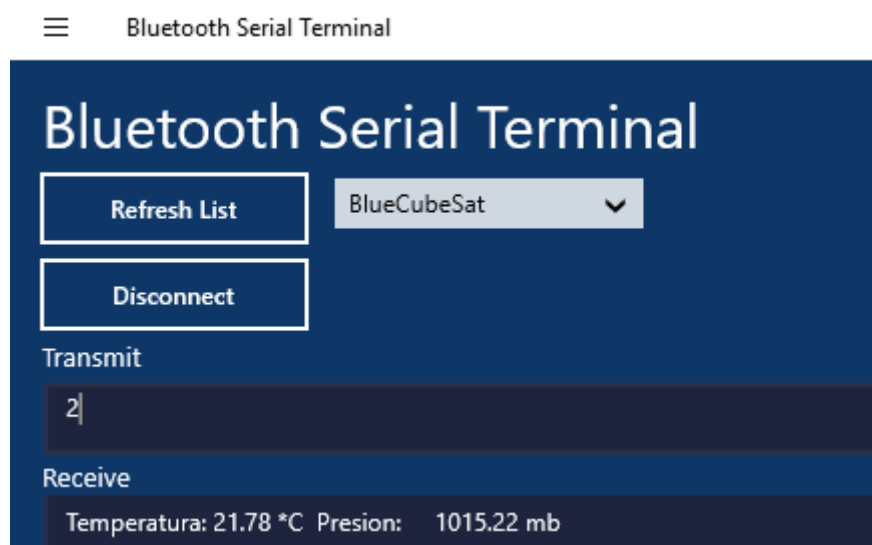


Imagen 27.- Lectura de datos en Bluetooth Serial Terminal del sensor barométrico BMP180.

Se ha comprobado la temperatura actual mediante 'Google' en Barcelona y corresponde a los datos enviados por el BMP180.



Imagen 28.- Temperatura Barcelona según la información de 'Google'

La presión al estar al nivel del mar, se ha obviado como correcta. El '2' que figura en la casilla de transmitir es el valor con el cual solicitamos los datos al módulo BMP180.

Para finalizar este apartado se ha colocado una imagen donde se puede observar como ha quedado el prototipo definitivo y funcional del sistema de comunicación basado en la protoboard, los componentes y Arduino Nano.

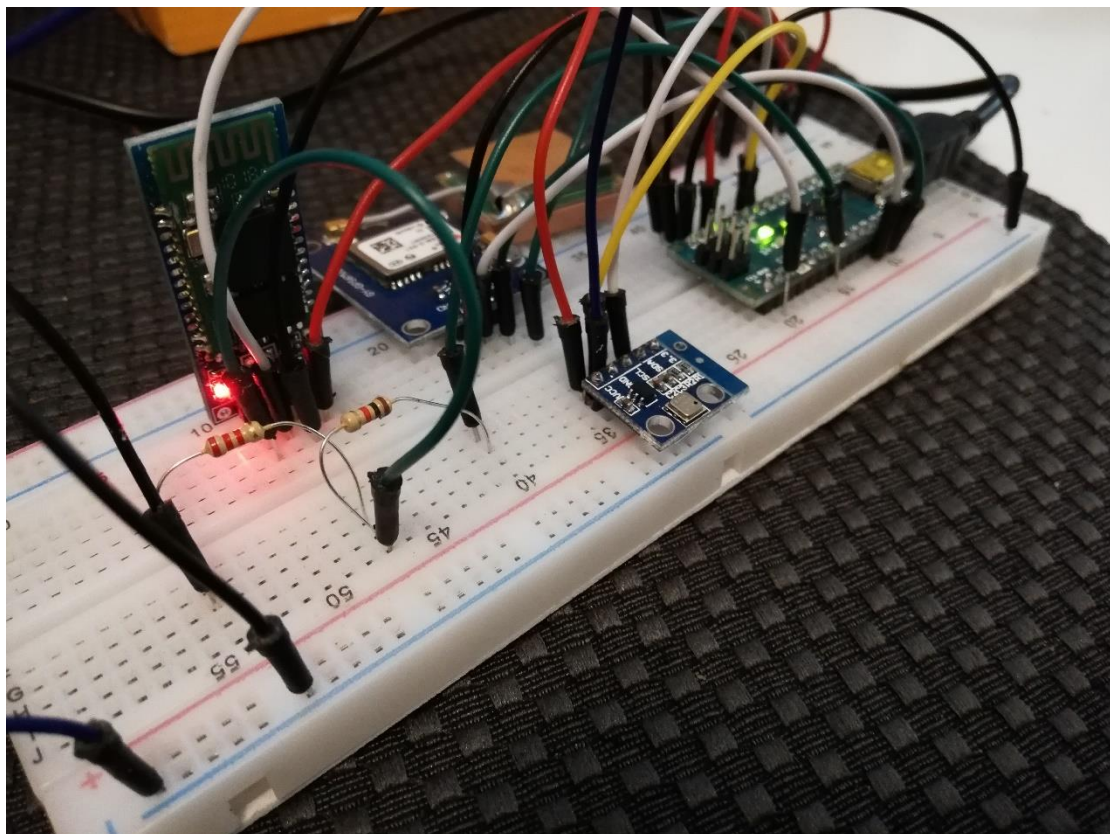


Imagen 29.- Prototipo de sistema de comunicación definitivo basado en la protoboard, Arduino y los módulos empleados

4. Diseño menú opciones de usuario

Cómo apunte final del capítulo para hacer una presentación de la lectura de datos interactiva con el usuario, se ha decidido crear un menú. En la siguiente figura se puede observar cómo se ha creado el menú en el entorno de programación Arduino.

```
void Menu() {
  BlueCubeSat.println("-----");
  BlueCubeSat.println("                      MENU  Lectura Datos CubeSat  UPC                      ");
  BlueCubeSat.println("-----");
  BlueCubeSat.println(" ");
  BlueCubeSat.println("          (1) LECTURA DATOS MODULO GPS");
  BlueCubeSat.println("          (2) LECTURA DATOS MODULO DE BMP180");
  BlueCubeSat.println(" ");
  varMenu=false;
}
```

Imagen 30.- Menú usuario

Este menú, permite al usuario introducir '1' para la obtención de los datos GPS y '2' para la obtención de los datos del sensor barométricos. A continuación, se ha colocado una muestra de cómo el usuario ve el menú desde una aplicación, en nuestro caso, desde Bluetooth Serial Terminal.

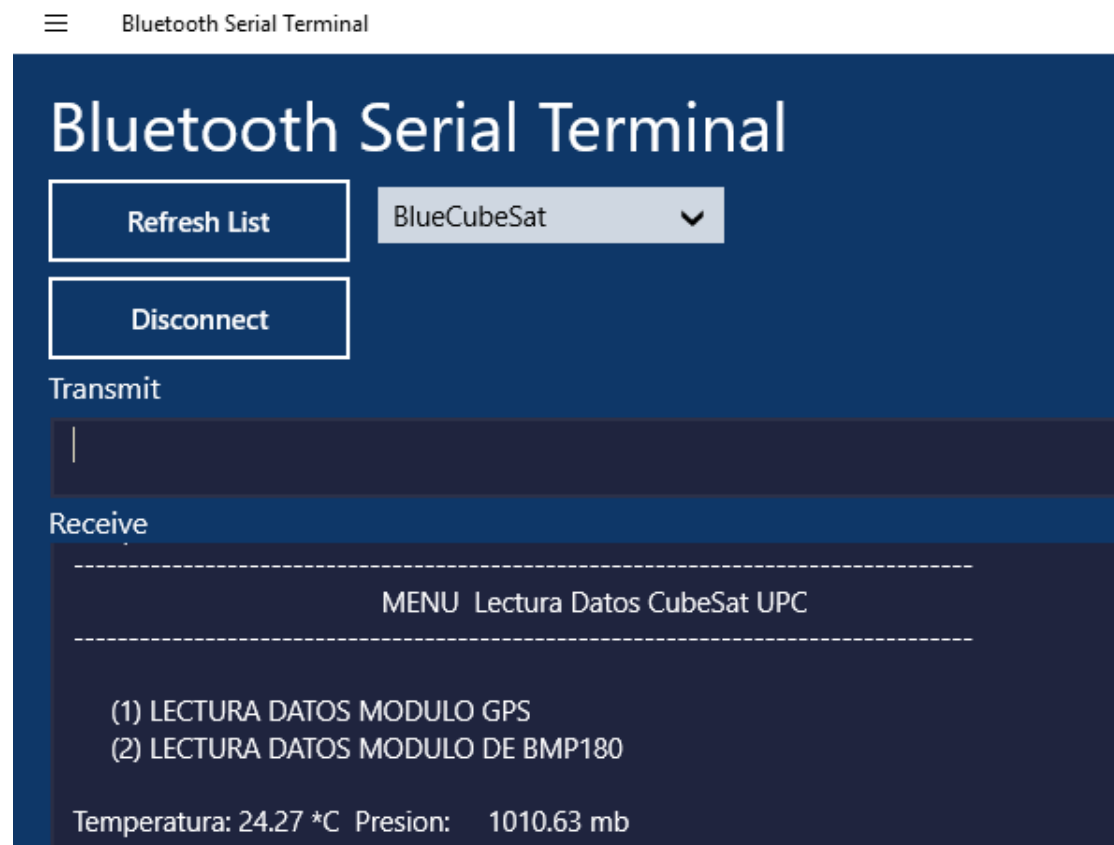


Imagen 31.- Menú Usuario Serial Terminal

Capítulo V. DISEÑO PCB

El objetivo de este capítulo es la creación de una placa de circuito impreso (printed circuit-Board, PCB) en el software KiCAD con la finalidad de eliminar el cableado y hacer un diseño compacto, adaptado a las medidas del CubeSat.

1. Introducción

KiCAD es un software de diseño *open-source* formado por varios programas que trabajan en conjunto para ayudar al usuario en el diseño de la PCB.

Los programas que componen KiCAD y que se han utilizado para la elaboración de la PCB son los siguientes:

1. *Schematic Capture*. En este programa diseñaremos el diagrama electrónico del circuito eléctrico realizado con la protoboard. En él incluiremos los diferentes componentes utilizados, las diferentes conexiones entre ellos y las conexiones a tierra y alimentación.

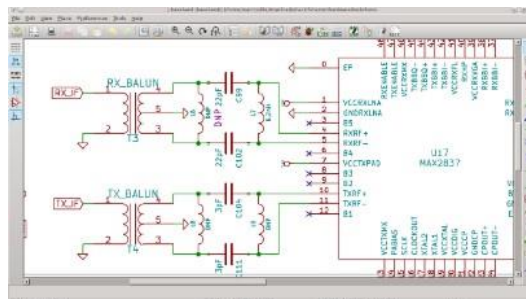


Imagen 32.- Esquemático KiCAD (source: <http://kicad-pcb.org/>)

2. *PCB Layout*. Programa para el trazado de pistas, perforaciones, dimensiones de la PCB y todos los datos relativos a la fabricación de la PCB. De aquí, saldrán los archivos definitivos que se enviarán al fabricante, los archivos GERBERS.

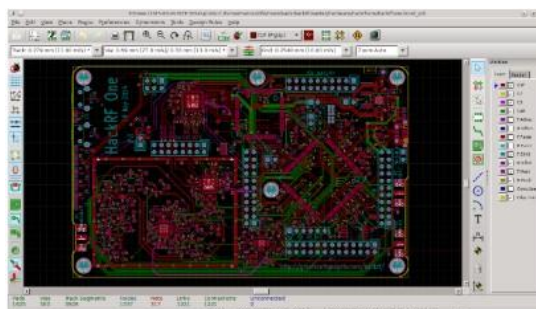


Imagen 33.- PCB Layout KiCAD (source: <http://kicad-pcb.org/>)

3. *3D Viewer*. Este programa permite hacer un seguimiento 3D de como está resultando la PCB.

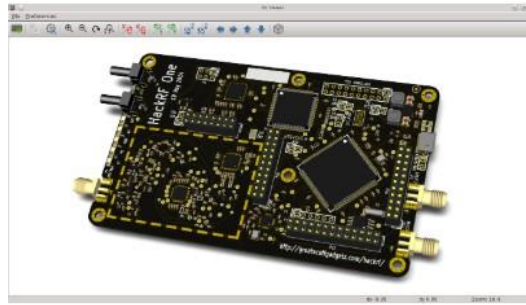
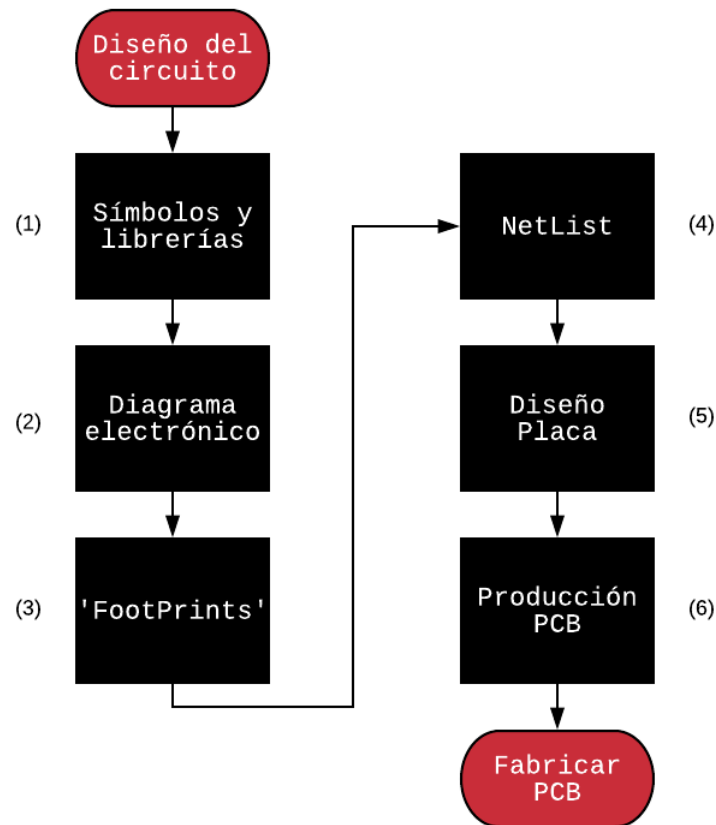


Imagen 34.- 3D Viewer KiCAD (source: <http://kicad-pcb.org/>)

2. Diagrama de proceso



- (1) Inserción de librería y símbolos necesarios para trabajar con el esquemático
- (2) Diseño del esquemático basándonos en el prototipo diseñado en el capítulo anterior
- (3) Conexión de las huellas o 'footprint' basándonos en los datasheet del fabricante
- (4) Asociamos los 'footprints' a los símbolos del esquemático
- (5) Diseñamos el trazado, perforaciones y configuramos las reglas de fabricación
- (6) Generamos los archivos de producción y visualizamos en 3D el diseño de la PCB.

3. Diseño

En las secciones siguientes se abordarán los pasos seguidos desde la realización del esquemático siguiendo el prototipo realizado en el capítulo anterior hasta la fabricación de la PCB.

3.1. SÍMBOLOS Y LIBRERÍAS

Antes de comenzar con el esquemático o diagrama eléctrico se deben incluir los símbolos y librerías que utilizaremos para este. KiCAD lleva por defecto incorporado algunos. Como es el caso del Arduino Nano que estamos utilizando para este proyecto. Los demás componentes debemos crearlos.

En la ventana principal de KiCAD nos dirigimos a: *Preferences >> Component Libraries*

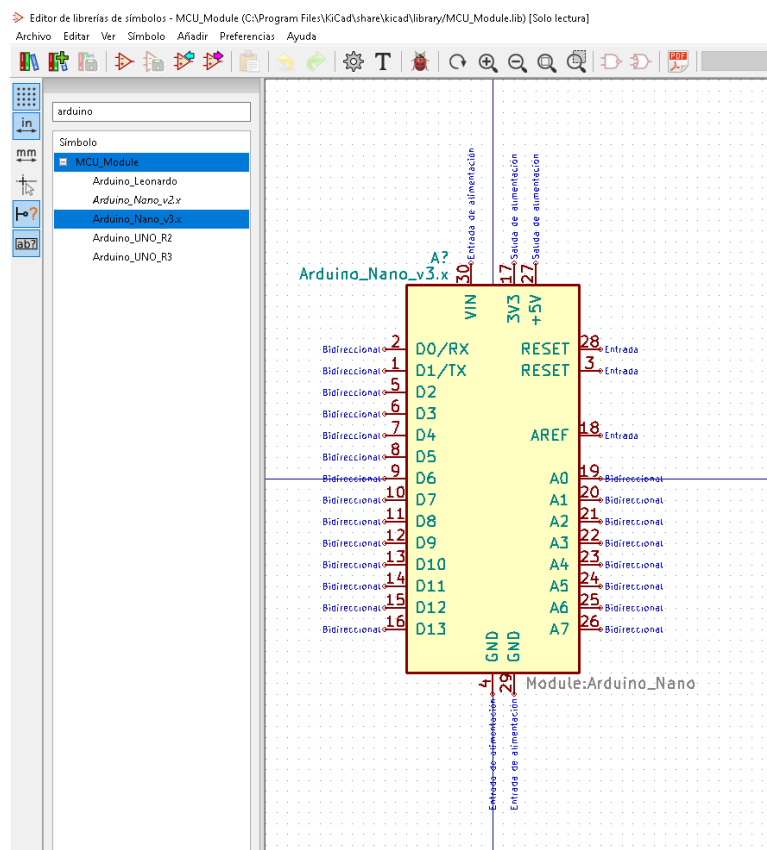


Imagen 35.- Arduino Nano Librería y footprint

Dado la complejidad de la creación de nuevos componentes desde cero en KiCAD se ha optado por simular los pines de los componentes usando módulos equivalentes al número de pads utilizados por el módulo en la conexión física con la protoboard, ya que el resultado es el mismo.

Por ejemplo, el módulo bluetooth consta de 32 pads (Anexo 4) de los cuales tan sólo se utilizan 4 en la conexión física con la protoboard, como se ha explicado en capítulos anteriores. En nuestro caso, se creará un componente de 4 pines que simulará ser el módulo Bluetooth.

Los 3 módulos creados para los componentes que se utilizarán (sensor, GPS y Bluetooth) se pueden observar en Imagen 37.

Un detalle importante es que para la recreación 3D de los componentes le asignaremos el objeto correspondiente 3D que conseguimos en la comunidad de www.grabcad.com.

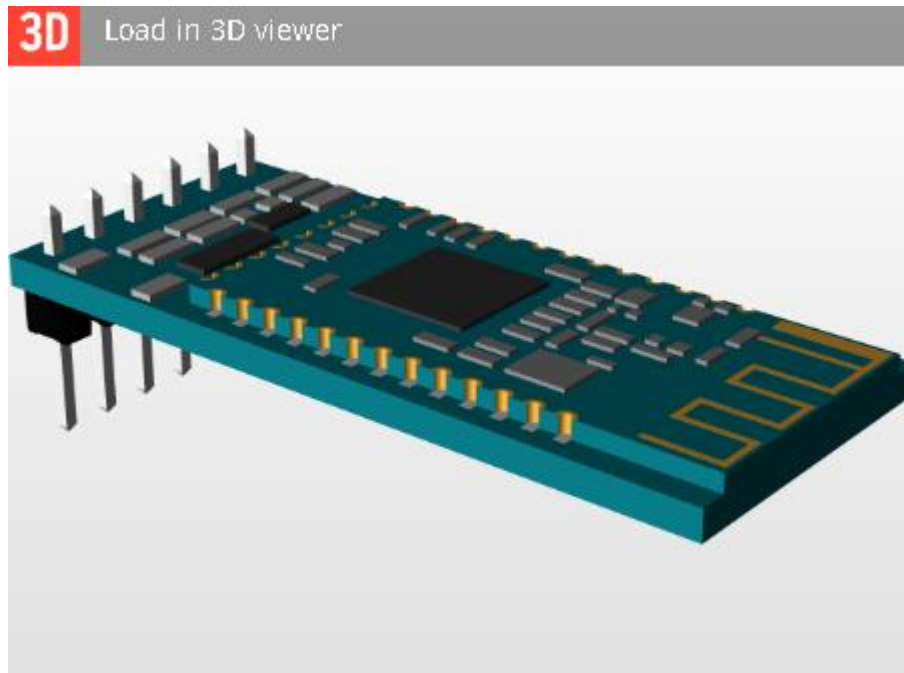


Imagen 36.- Ejemplo de módulo Bluetooth para recreación 3D

Al no realizar los componentes, no tendremos la dimensión real de los mismos. Por eso, la visualización 3D es de gran ayuda para la distribución de estos en la PCB. Evitando crear superposiciones de los mismos.

3.2. DIAGRAMA ELECTRÓNICO

Con todas las librerías cargadas, se procede a la colocación de los componentes en el esquemático y su conexionado. Este paso es importante debido a que se le informa a KiCAD cómo están relacionados los componentes basándonos en las conexiones que hemos realizado físicamente del Arduino Nano con sus componentes en el capítulo anterior. En La figura siguiente se observa cómo queda el diagrama electrónico.

En la ventana principal de KiCAD nos dirigimos a: *Eeschema*





Diagram illustrating the PCB layout for the Raspberry Pi 4B, showing component placement and dimensions.

Dimensions:

- Vertical distance between the top and bottom component rows: 27,94 mm
- Vertical distance between the bottom component row and the bottom edge: 10,16 mm
- Horizontal distance between the component rows: 10,16 mm

Component Placement and Pin Connections:

Component	Pin
NC	NC
NC	NC
SCK	MOSI
MISO	SS
TX	RX
SCL	SDA
GND	5 V
GND	3,3 V
GND	9,6 V

Imagen 38.- Distribución sockets 'qbkkit' de 'Open Cosmos'

PCB footprint

3.4. NETLIST

Netlist es un archivo que le informa a KiCAD como debe estar conectado cada terminal de cada componente.

Para acceder a *Netlist* en la ventana principal de *Eeschematics*, clicamos sobre *Generate Netlist*.



No todos los componentes están asociados con sus huellas por defecto. En este paso debemos asociar aquellos que no lo están antes de generar el archivo *Netlist*.

Para ello, simplemente escogemos el componente a asociar y buscamos la huella en la librería disponible. El resultado de esta asociación es el siguiente:

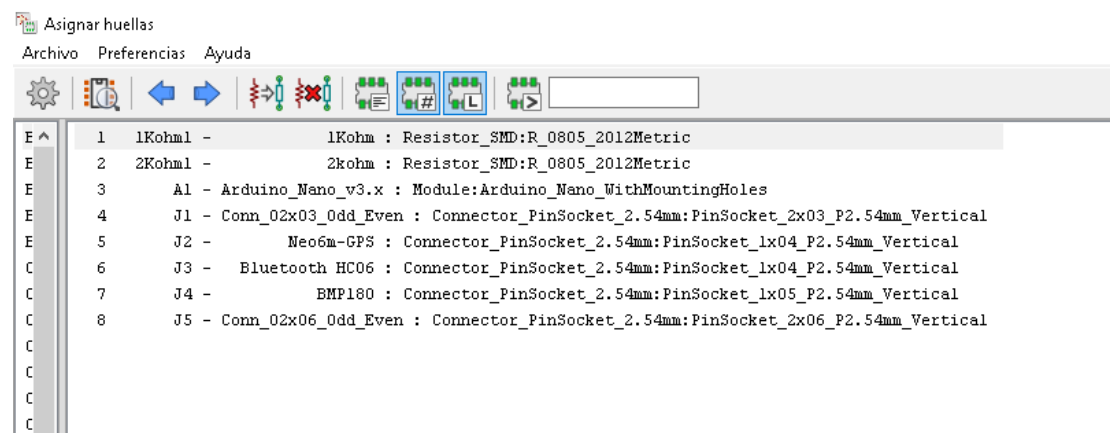


Imagen 39.- *Netlist*. Asociación de huellas con componentes

3.5. DISEÑO PLACA

Todos lo generado desde el principio del proyecto hasta ahora, queda resumido al archivo *Netlist*. El archivo que procesa el programa PCB Layout.

En la ventana principal de KiCad, clicamos sobre: *Pcbnew*.



Cuando se carga el archivo el programa automáticamente genera las conexiones que se le ha indicado en la sección anterior. Pero, debemos indicarle los trazados de las vías, las perforaciones y las propiedades de la placa, basándonos en las dimensiones del CubeSat del laboratorio y aplicando las reglas generales de las PCB, adjuntas en el Anexo 1.

Una vez realizado el trazado de todas las vías y las dimensiones de la placa, la PCB resultante sería la siguiente:

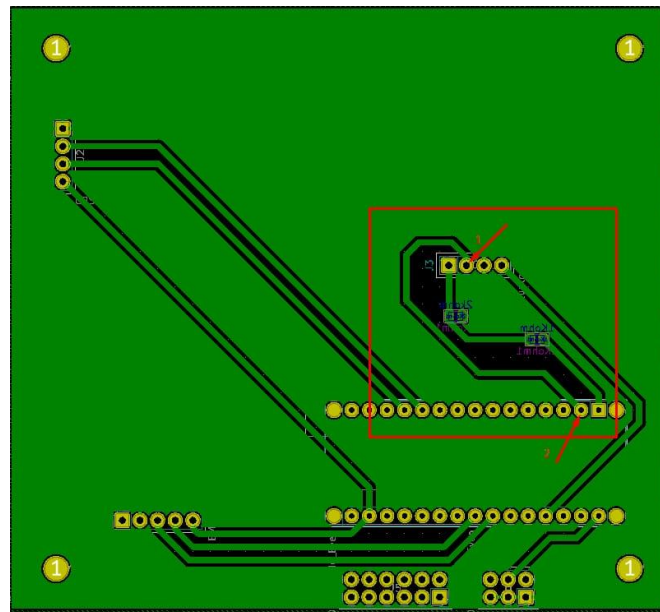


Imagen 40.- PCB Version 1

Como vemos en el recuadro rojo marcado en la imagen anterior, esa zona puede darnos conflictos. Debido a que puede provocar una acumulación de tensión y, esto, podría generar daños o fallos en el circuito. Por este motivo, se ha decidido crear una puente o grapa desde el punto 1 al dos para que esta zona este en contacto con la masa de la placa.

El resultado final de lo aplicado anteriormente, se resume en la siguiente imagen, siendo esta la placa definitiva que se enviará a fabricar.

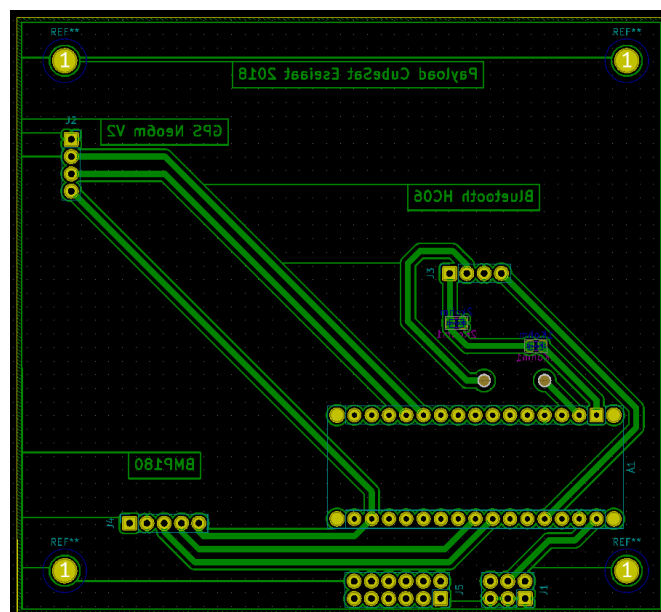


Imagen 41.- Diseño final PCB

3.6. PRODUCCIÓN PCB

Para la producción de la PCB se han de generar los archivos Gerber. Hay una función en el software KiCAD para esta opción. Dado que cada fabricante tiene sus propias reglas de fabricación, a la hora del diseño, se ha realizado con las reglas de aisler.net con el fin de pedir la placa en esa compañía.

Accediendo a la web y cargando los archivos GERBER se podrá tener una visión final de la producción de la placa.

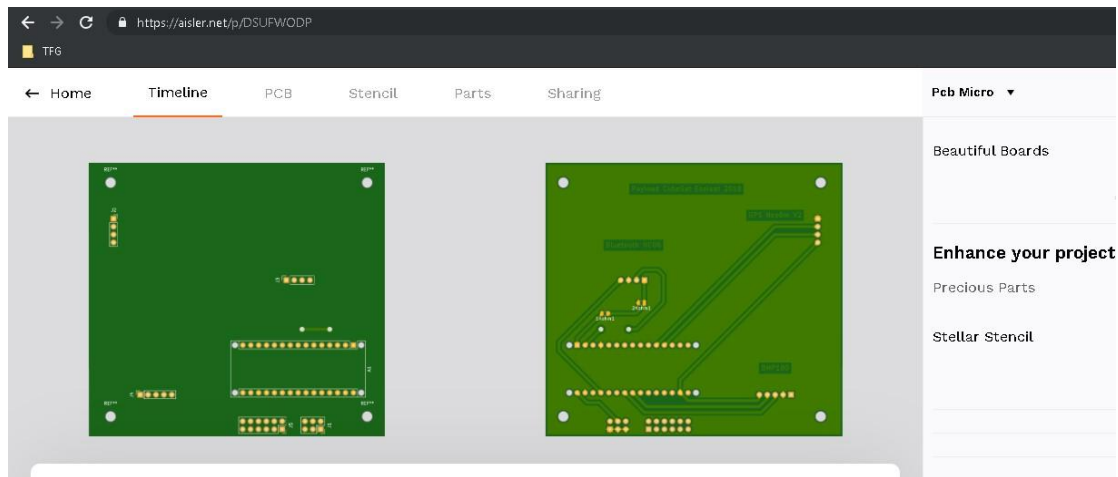


Imagen 42.- Vista previa a la fabricación de la PCB en Aisler.net

3.7. VISTA 3D

A continuació, se presenta el resultat 3D de tots els passos realitzats en les seccions anteriors.

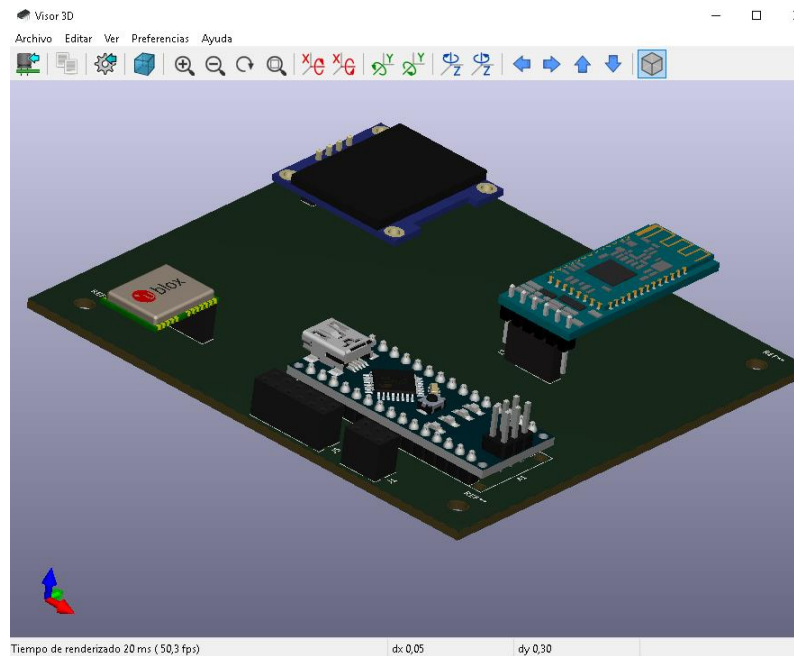


Imagen 43.- Vista posterior PCB en 3D Viewer KiCAD

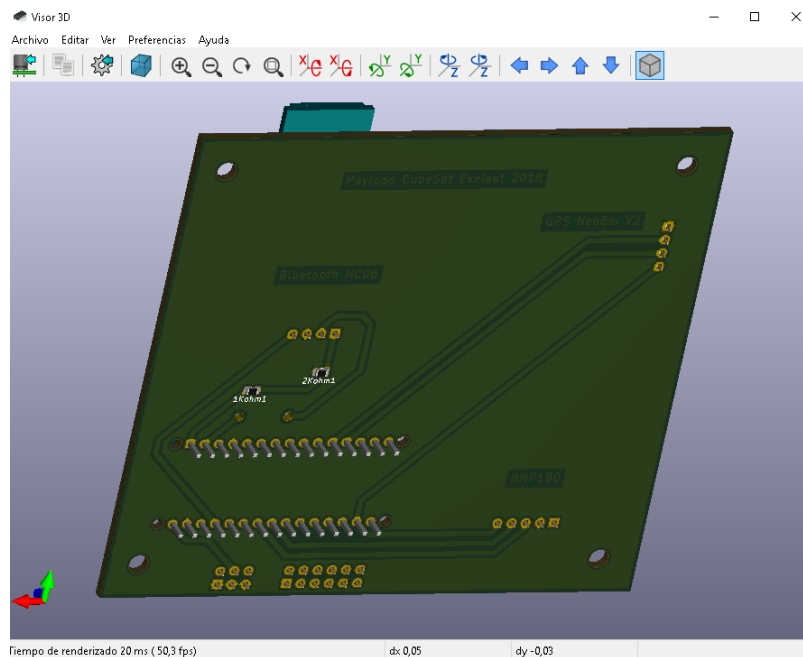


Imagen 44.- Imagen 23.- Vista inferior PCB en 3D Viewer KiCAD

Capítulo VI. IMPLEMENTACIÓN EN EL CUBESAT

En este capítulo se realizará un análisis de la placa recibida por el fabricante, las modificaciones que han sido necesarias para mejorar la implementación en el CubeSat, para así obtener los resultados deseados.

1. Placa PCB recibida

En la siguiente imagen se puede observar la placa recibida:

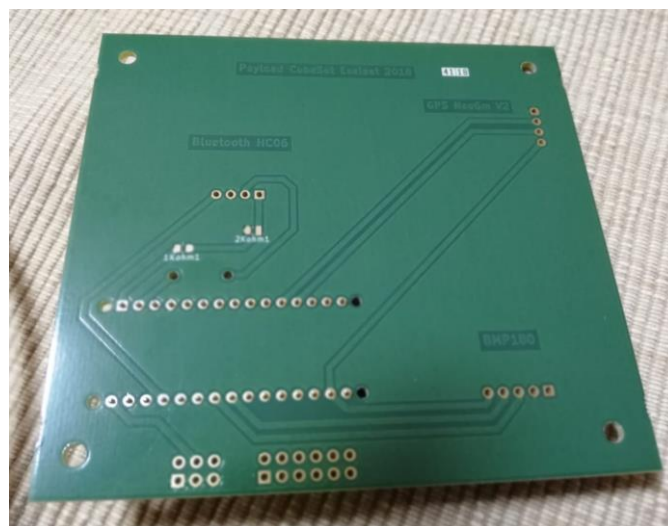


Imagen 45.- Placa PCB recibida pendiente de soldar componentes y de modificaciones

El dimensionado de la placa y el diseño de las vías y las perforaciones es acorde al solicitado. Antes de comenzar a realizar la colocación de los componentes del proyecto, será necesario comprobar el correcto funcionamiento de las vías. Para ello se ha procedido con un Tester a probar las conexiones entre punto y punto y, se ha verificado, que todos los puntos están correctamente comunicados.

Se ha decidido, pensando en futuras modificaciones, actualizaciones o cambios, que los componentes no sean soldados directamente en las placas, sino, sean colocados sobre pines hembra.

Por ello, como primer paso se han soldado todos los pines hembras en la parte posterior de la PCB, donde se colocarán los componentes. Como se observa en la imagen de a continuación:

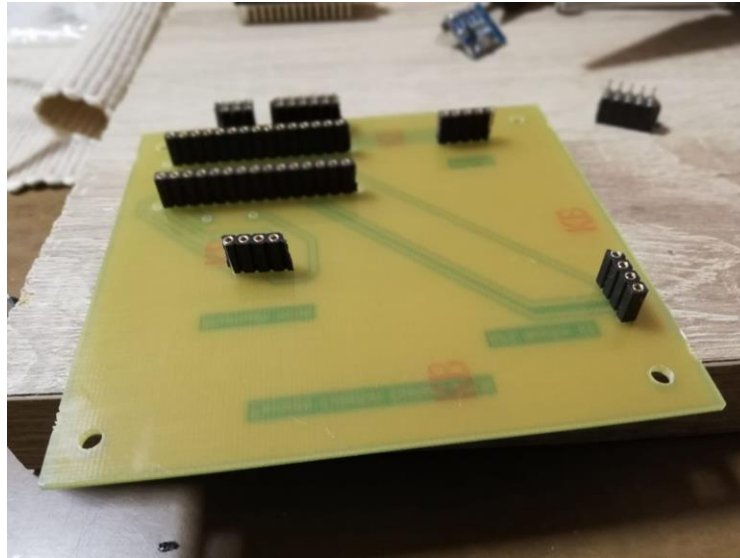


Imagen 46.- Soldado de pines de la PCB

El diseño de la placa incluía dos resistencias y un puente. Por ello, será necesario soldar estos.

Las resistencias SMD no se han podido encontrar en distribuidores locales, por tanto, se ha procedido a desoldarlas de placas antiguas y soldarlas en la PCB del presente proyecto. Antes de este paso se ha verificado con el Multímetro que estas resistencias estuvieran aún operativas.

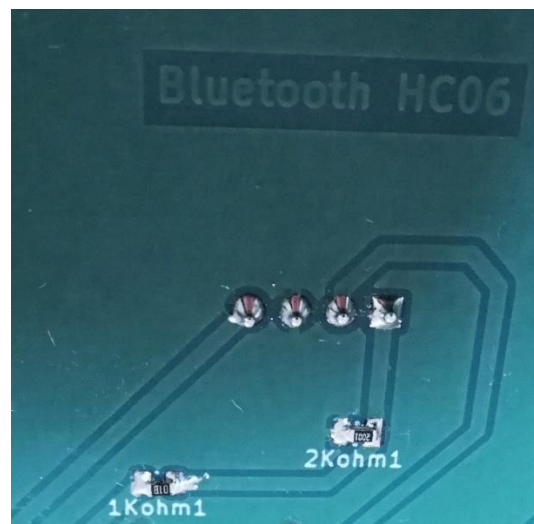


Imagen 47.- Resistencias SMD

En la realización del puente se han tenido que limar las perforaciones con tal de poder soldar la grapa, debido a que, la placa no traía los pads. Una vez limado, se ha soldado una grapa por la capa posterior de la PCB.

2. Modificaciones

Se ha aplicado una única modificación a la PCB debido a un fallo del conector 3.3 del BMP180. El conector 3.3 no proporciona alimentación al módulo BMP180. Por tanto, al conectar todos los componentes, el módulo no respondía. Para solventar el problema se ha colocado un puente entre el punto 3.3 y el VCC como se observa en la imagen siguiente:

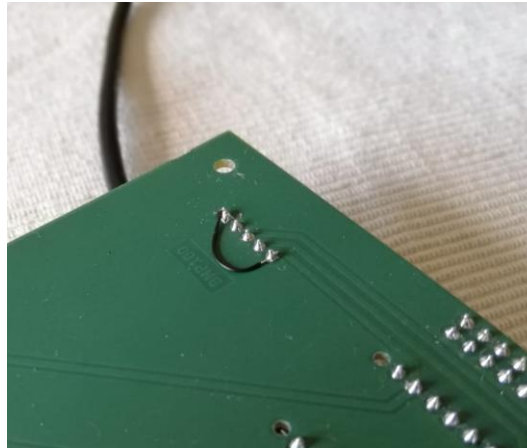


Imagen 48.- Puente para corregir error de conexionado

Una vez hechas las modificaciones y corregido los fallos ya estaría disponible el diseño final de la PCB:

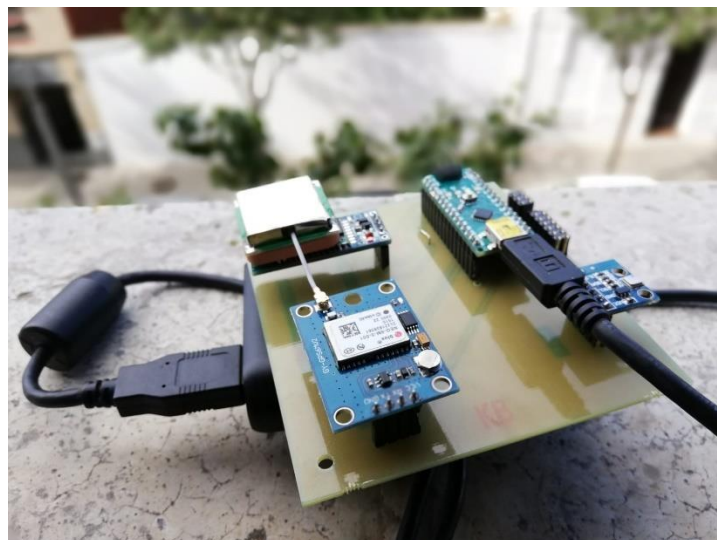


Imagen 49.- Diseño Final PCB.

3. Implementación en CubeSat

Una vez hecho los ajustes finales y la integración de los componentes con la PCB se obtiene la lectura de los datos del módulo GPS Neo-6mV2 y BMP180 a través de la conexión inalámbrica bluetooth y mediante la aplicación 'Bluetooth Serial Terminal'



Imagen 50.- Test final lectura de datos con PCB integrada en CubeSat

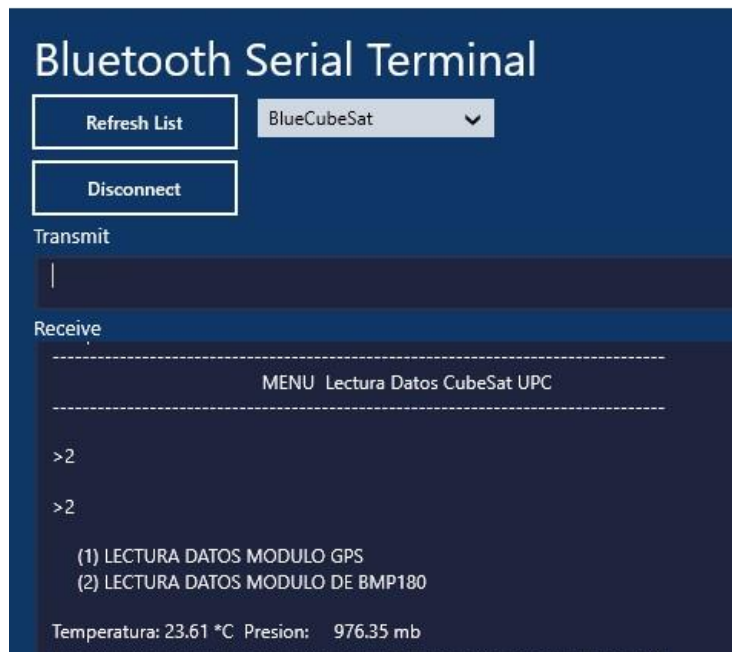


Imagen 51.- Comunicación final de lectura de datos

La evaluación de los datos recibidos corresponde a la temperatura, presión y ubicación actual de donde se encuentra el CubeSat.

Por otra parte, como se observa a continuación, el ensamblaje de la PCB con el CubeSat es el adecuado. Como también lo es, el amoldamiento con los pines que permiten la transmisión de datos y alimentación entre capas del 'qbkkit'.

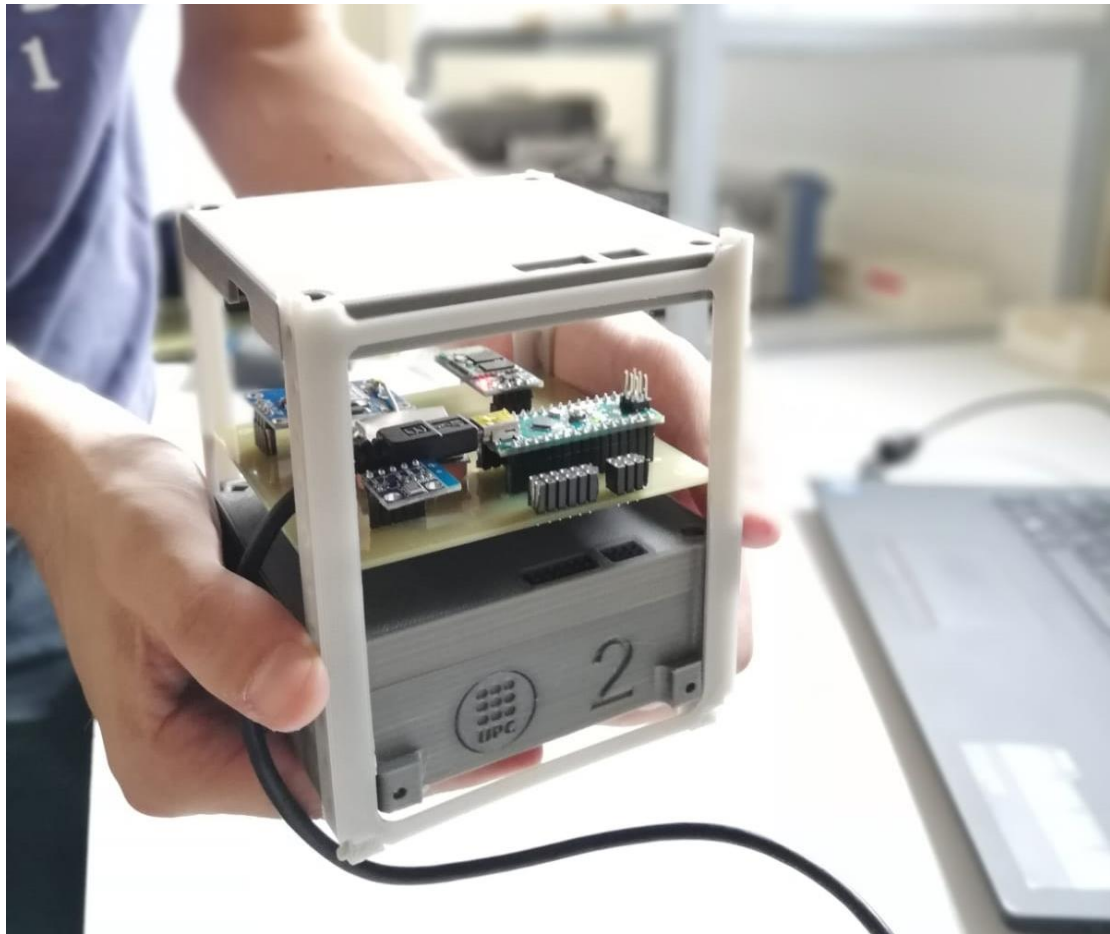


Imagen 52.- Diseño Final de PCB implementada en CubeSat

Capítulo VII. CONCLUSIONES

Se ha conseguido el alcance establecido obteniendo resultados positivos para cada uno de los objetivos planteados en un principio. Ya que se ha podido demostrar el rendimiento del desarrollo de un sistema de comunicación que transmite y recibe datos vía comunicación 'wireless' en tiempo real.

Por el lado de programación, después de trabajar con el software 'open source' Arduino, considero que es una herramienta muy útil para primeras aproximaciones al mundo de la electrónica y programación, pero a la hora de comenzar a realizar tareas más específicas de programación, hay una escasez de documentación avanzada, lo que aumenta el tiempo de trabajo en proyectos más complejos.

En este proyecto la programación de Arduino ha cumplido con todas las expectativas, aunque, debido a lo comentado en el apartado anterior, el desarrollo del backend del sistema de comunicaciones ha tomado más tiempo del previsto.

Respecto al diseño de la PCB, es una tarea compleja que requiere de una curva de aprendizaje lenta al principio. Debido a que es necesario una desenvoltura en tres diferentes programas con multitud de funciones, librerías, diseño 3D, entre otros; además de la teoría que hay detrás de estos. Por consiguiente, considero que se necesita una preparación previa para desarrollar este tipo de prototipos.

Por otra parte, una vez desarrollado todo el sistema de comunicación e implementado en el CubeSat se puede apreciar el potencial de la miniaturización de la electrónica. Ya que las nuevas herramientas tecnológicas junto con el Internet, ponen al alcance de muchos usuarios, conseguir dispositivos de pequeño tamaño con prestaciones similares a las de sus semejantes de grandes tamaños, además de reducir el coste económico, inclusive herramientas de libre acceso como lo son los softwares utilizados en este proyecto.

Se deja para futuros proyectos la realización de diferentes tareas que aumentarían las prestaciones del diseño de este prototipo como; la comunicación bidireccional entre dos CubeSats compartiendo sus ubicaciones y sus datos barométricos, con el servicio de una base de datos, para almacenar y realizar posteriores estudios y seguimientos de estos. Y la sustitución del módulo bluetooth por un sistema de transmisión UHF (Ultra High Frequency) que es la banda libre de radioaficionados que se usa en aplicaciones de satélite de bajo coste como el CubeSat.

PRESUPUESTO

El presupuesto del proyecto se ha realizado teniendo en cuenta las diferentes áreas que conlleva, desde el trabajo de programación, pasando por el software necesario, hasta los elementos utilizados para su desarrollo. Para la remuneración económica de mano de obra, se ha estipulado el salario medio de un estudiante de ingeniería en prácticas de la UPC.

Tabla III.- Presupuesto del proyecto

Ingeniería y software de proyecto			
Descripción	Horas laborables	Precio €/hora	Total
Programador Backend	84	8	672
Diseño PCB	144	8	1152
Unidad de Testing	48	8	384
Documentación	190	8	1520
Subtotal	466		3728
Software utilizado			
Descripción	Precio unidad	Cantidad	Total
Arduino	Open source	1	
KiCAD	Open source	1	
Fritzing	Open source	1	
Hardware utilizado			
Descripción	Precio unidad €	Cantidad	Total
Arduino nano	24.42	1	24.42
Módulo BMP 180	4.3	1	4.3
GPS Neo 6MV2	19.6	1	19.6
Bluetooth HC06	6.2	1	6.2
Printed Circuit Board	87.49	1	87.49
Cableado	4.36	1	4.36
Protoboard	4.99	1	4.99
Subtotal	151.36		151.36
Precio total del proyecto			3879.36

BIBLIOGRAFÍA

- [1] Mendoza, Mariale, "CubeSat Design Specification Rev. 13 The CubeSat Program, Cal Poly SLO CubeSat Design Specification (CDS) REV 13 Document Classification X Public Domain ITAR Controlled Internal Only." .
- [2] Flores Salado, Guadalupe, Microválvula de bajo consumo para plataformas microfluídicas en PCB [Recurso electrónico] : Trabajo Fin de Máster "CAPÍTULO 2. ESTADO DEL ARTE." Available: <http://bibing.us.es/proyectos/abreproy/70482/>
- [3] NASA, "Small Spacecraft Technology State of the Art" no. February, 2014. Available: https://www.nasa.gov/sites/default/files/atoms/files/small_spacecraft_technology_state_of_the_art_2015_tagged.pdf
- [4] "KickSat Sprites set for launch March 30 | AMSAT-UK." [Online]. Available: <https://amsat-uk.org/2014/03/13/kicksat-sprites-set-for-launch/>. [Accessed: 15-Mar-2018].
- [5] J. Torres Riera, "NANOSATÉLITES, MICROSATÉLITES Y MINISATÉLITES: UNA ALTERNATIVA DE FUTURO." Available: http://rseap.webs.upv.es/Anales/06/A_Nanosatelites.pdf
- [6] "Definición de circuito integrado - Qué es, Significado y Concepto." [Online]. Available: <https://definicion.de/circuito-integrado/>. [Accessed: 15-Mar-2018].
- [7] Bethesda MD (SPX), "History of the CubeSat," *Bethesda MD (SPX)*, 2016. [Online]. Available: http://www.spacedaily.com/reports/History_of_the_CubeSat_999.html. [Accessed: 20-Mar-2018].
- [8] I. Ibáñez, "¡Luces, cámaras y cubesats! - Opinión Infoespacial," 2017. [Online]. Available: <http://www.infoespacial.com/ie/2017/10/04/opinion-luces-cameras-cubesats.php>. [Accessed: 20-Mar-2018].
- [9] "Planet, Analytics." [online]. Available: <https://www.planet.com/products/analytics/>. [Accessed: 20-Mar-2018].
- [10] "AIS - Karten Space." [Online]. Available: <http://kartenspace.com/ais/>. [Accessed: 20-Mar-2018].
- [11] "3-Diamonds Sky and Space Global." [Online]. Available: <https://www.skyandspace.global/operations-overview/technology/>. [Accessed: 20-Mar-2018].
- [12] "Modulo Bluetooth HC-06 | Blog de PatagoniaTec Electronica." [Online]. Available: <https://saber.patagoniatec.com/2014/06/modulo-bluetooth-hc-06-bluetooth-arduino-slave-hc06-esclavo-iot/>. [Accessed: 27-Sep-2018].
- [13] Morales. Michel, "Protocolo I2C - Fundamentos de aprendizaje. ¿Cómo funciona? - Clases TBem.," 2017. [Online]. Available: <https://teslabem.com/learn/fundamentos-del-protocolo-i2c-aprende/>. [Accessed: 09-Sep-2018].
- [14] M. J. B. Díaz, "Normas Básicas y Recomendaciones en el Diseño de PCBs," pp. 1–20, 2015. Available: <https://www.dte.us.es/docencia/etsii/gii-ic/laboratorio-de-desarrollo-hardware/temas/Tema5NormasPCB/view>

- [15] Bosch Sensortec, "BMP180 Digital pressure sensor," no. April, p. 28, 2013.
- [16] L. Guangzhou HC Information Technology Co ., "HC06 Datasheet," *GuangZhou, China*, no. 13, pp. 1–17, 2011.
- [17] U-blox, "NEO-6 GPS Modules Data Sheet," p. 25, 2011.
- [18] "Arduino Nano (V2.3) User Manual Released under the Creative Commons Attribution Share-Alike 2.5 License."

ANEXOS

1. Normas y recomendaciones en el layout de la PCB

Recomendaciones generales sobre el Layout [14]

- Tener claro el grid (cuadrícula) que se va a emplear en la colocación de los componentes y en el trazado de las pistas o nets. Las herramientas suelen traer un grid por defecto de 50mil (0,05inch) y un valor alterno menor, 25mils (se emplea con la tecla alt). Estos valores suelen ser adecuados con la mayoría de los componentes
- Si se posee limitaciones en cuanto al tamaño de la PCB lo primero que se debe hacer es ajustar el área de PCB al tamaño máximo permitido
- Realizar la colocación de componentes, comenzando por los componentes de entrada/salida. Estos deben colocarse de acuerdo al diseño que se tenga en mente sobre cómo debe quedar dispuestos los mismos.
- Para el resto de componentes, se debe seguir el criterio de agruparlos según bloques lógicos. De esta manera se minimizarán las trazas de conexión. Siempre debe intentarse hacer las pistas más cortas posibles minimizando así efectos parásitos resistivos y capacitivos

Recomendaciones y normas sobre el trazado de pistas

- Seleccionar el ancho de pista que vaya a emplearse. El ancho de la pista debe ser suficiente para permitir el paso de la corriente máxima que vaya a circular por la pista.
- En las PCBs habituales con ancho de cobre de 35micras:

Ancho pista (mm)	Corriente máxima (A)
4	10
2	5
1,5	4
1	3

0,5	2
0,2	0.5

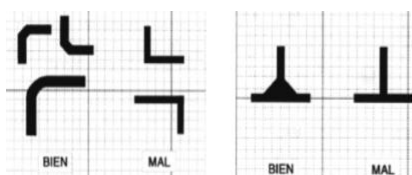
- En las señales digitales rara vez se superan corrientes máximas de 500mA (ejemplos: pines de salida Arduino: máxima corriente 40mA, raspberry Pi: 16mA, motor de DC de 9volt, máxima corriente consumida 500mA)

Recomendaciones y normas sobre el trazado de pistas

- La separación mínima entre pistas depende de la tensión que soporten.
- Para tensiones de trabajo digitales (5v-10v) una separación mínima de 0,3mm es suficiente.

Recomendaciones y normas sobre el trazado de pistas

- En el trazado de las pistas es conveniente evitar el trazado de ángulos de 90°.



Recomendaciones y normas sobre el trazado de pistas

- La distancia mínima entre pistas y los bordes de la placa será de dos décimas de pulgada, aproximadamente unos 5 mm.
- No se pasarán pistas entre dos terminales de componentes activos (transistores, tiristores, etc.). Si se puede hacer entre terminales de componentes pasivos
- Se debe prever la sujeción de la placa a un chasis o caja; para ello se dispondrá un taladro de 3,5 mm en cada esquina de la placa.
- Cuando dos o más pistas discurren paralelas debería mantenerse la distancia de separación



2. Código sistema de comunicaciones Arduino

```

1 //Configuración módulo Bluetooth HC-06 + GPS Neo 6m
2 //Guillermo Nicolás Bustamante
3 //TFG - Octubre 2018
4
5 #include <SoftwareSerial.h>           // Incluimos la librería SoftwareSerial
6                                     // que nos permite ampliar el número de puertos
7                                     // para la comunicación en serie
8
9
10 #define BlueCubeSat Serial           //Cambiamos el nombre del puerto serie por 'BlueCubeSat'
11 SoftwareSerial GpsCubeSat(8,9);      // Definimos los pines 8 y 9 como RX y TX
12
13 #include <NMEAGPS.h>                 // Librería para parsear la información que
14                                     // recibimos del GPS
15
16 #include <SFE_BMP180.h>              // Incluimos librería de SparkFun
17 SFE_BMP180 bmp180;                  // Creamos un objeto de la clase SFE_BMP180
18
19
20 NMEAGPS GpsObject;                  // Creamos un objeto de la clase NMEAGPS y lo llamamos 'GpsObject'
21 gps_fix data;                       // Creamos una variable que contendrá todos los datos que obtengamos del gps
22                                     // y la llamamos 'data'
23
24 int varMenu=true;                   // Creamos una variables para crear el menu
25
26 void setup(){
27
28     BlueCubeSat.begin(9600);         // Inicializamos el puerto serie BlueCubeSat que hemos creado
29     while(!BlueCubeSat);             // Esperamos a que inicie
30
31     GpsCubeSat.begin(9600);          // Inicializamos el puerto serie del Gps
32     while(!GpsCubeSat);              // Esperamos a que inicie
33     bmp180.begin();                  // Iniciamos BMP180
34 }
35
36 void loop(){
37
38     if (!varMenu){                   //Creamos un bucle para el menu
39         Menu();
40         varMenu=true;
41     }
42
43     if (GpsObject.available(GpsCubeSat)) { // Comprueba que el puerto Gps este operativo
44         data = GpsObject.read();        // Lee todos los datos del gps y los registra en data
45     }
46
47     if(BlueCubeSat.available())       // Comprueba si ha llegado algún dato por el puerto BlueCubeSat
48     {
49         char BluetoothData=BlueCubeSat.read(); // Guardamos el dato recibido en una variable
50         varMenu=false;
51         if (BluetoothData=='1'){
52             // Comprueba si el usuario ha presionado '1'
53             BlueCubeSat.println("Satellites: (u)");
54             BlueCubeSat.println( data.satellites );           // Número de Satélites que ha localizado en órbita
55             BlueCubeSat.println("Latitud:");
56             BlueCubeSat.println( data.latitude(), 6);         // Latitud de ubicación de CubeSat

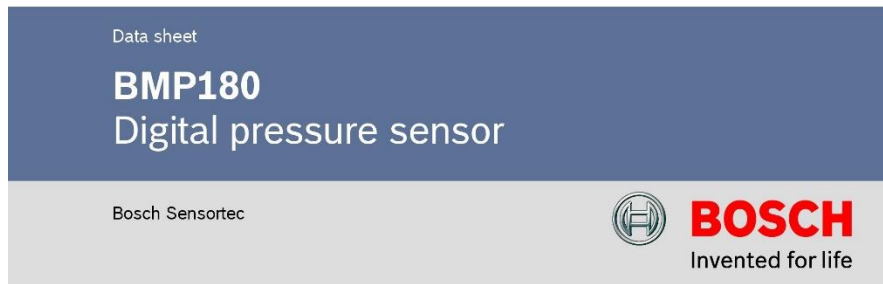
```

```

57 BlueCubeSat.println("Longitud:");
58 BlueCubeSat.println( data.longitude(), 6); // Longitud de ubicación de Cubesat
59 BlueCubeSat.println("Velocidad (m/s):");
60 BlueCubeSat.println( data.speed_mph()*0.44704 ); // Velocidad del CubeSat. Multiplicamos por 0.44704 para
61 BlueCubeSat.println("Altura: (m)");
62 BlueCubeSat.println( data.altitude()*0.3048); // Altura del Cubesat.
63 BlueCubeSat.println();
64
65 }
66
67 if (BluetoothData=='2'){ // Si presionamos 2 ...
68     char status;
69     double T,P;
70     status = bmp180.startTemperature(); //Inicio de lectura de temperatura
71     if (status != 0){
72         delay(status); //Pausa para que finalice la lectura
73         status = bmp180.getTemperature(T); //Obtener la temperatura
74         if (status != 0){
75             status = bmp180.startPressure(3); //Inicio lectura de presión
76
77             if (status != 0){
78                 delay(status); //Pausa para que finalice la lectura
79                 status = bmp180.getPressure(P,T); //Obtenemos la presión
80
81                 if (status != 0){
82                     Serial.print("Temperatura: ");
83                     Serial.print(T,2);
84                     Serial.print(" *C ");
85                     Serial.print(" ");
86
87                     Serial.print("Presion: ");
88                     Serial.print(P,2);
89                     Serial.println(" mb");
90                 }
91             }
92         }
93     }
94 }
95 }
96
97 void Menu(){
98     BlueCubeSat.println("-----");
99     BlueCubeSat.println("MENU Lectura Datos CubeSat UPC");
100    BlueCubeSat.println("-----");
101    BlueCubeSat.println("");
102    BlueCubeSat.println(" (1) LECTURA DATOS MODULO GPS");
103    BlueCubeSat.println(" (2) LECTURA DATOS MODULO DE BMP180");
104    BlueCubeSat.println("");
105    varMenu=false;
106 }
107

```

3. BOSCH BMP 180 Digital Pressure Sensor – DATASHEET [15]

**BMP180 Data sheet**


Document revision 2.5

Document release date 5 April 2013

Document number BST-BMP180-DS000-09

Technical reference code(s) 0 273 300 244

Notes Data in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product's appearance.

 BOSCH	Data sheet BMP180	Page 2
--	-----------------------------	--------

BMP180


DIGITAL PRESSURE SENSOR

Key features

Pressure range:	300 ... 1100hPa (+9000m ... -500m relating to sea level)
Supply voltage:	1.8 ... 3.6V (V _{DD}) 1.62V ... 3.6V (V _{DDIO})
Package:	LGA package with metal lid Small footprint: 3.6mm x 3.8mm Super-flat: 0.93mm height
Low power:	5µA at 1 sample / sec. in standard mode
Low noise:	0.06hPa (0.5m) in ultra low power mode 0.02hPa (0.17m) advanced resolution mode
<ul style="list-style-type: none"> - Temperature measurement included - I²C interface - Fully calibrated - Pb-free, halogen-free and RoHS compliant, - MSL 1 	

Typical applications

- Enhancement of GPS navigation (dead-reckoning, slope detection, etc.)
- In- and out-door navigation
- Leisure and sports
- Weather forecast
- Vertical velocity indication (rise/sink speed)

 BOSCH	Data sheet BMP180	Page 3
--	-----------------------------	--------

BMP180 general description


The BMP180 is the function compatible successor of the BMP085, a new generation of high precision digital pressure sensors for consumer applications.

The ultra-low power, low voltage electronics of the BMP180 is optimized for use in mobile phones, PDAs, GPS navigation devices and outdoor equipment. With a low altitude noise of merely 0.25m at fast conversion time, the BMP180 offers superior performance. The I²C interface allows for easy system integration with a microcontroller.

The BMP180 is based on piezo-resistive technology for EMC robustness, high accuracy and linearity as well as long term stability.

Robert Bosch is the world market leader for pressure sensors in automotive applications. Based on the experience of over 400 million pressure sensors in the field, the BMP180 continues a new generation of micro-machined pressure sensors.



 BOSCH	Data sheet BMP180	Page 4
--	-----------------------------	--------

Index of Contents


1. ELECTRICAL CHARACTERISTICS	6
2. ABSOLUTE MAXIMUM RATINGS.....	8
3. OPERATION	9
3.1 GENERAL DESCRIPTION	9
3.2 GENERAL FUNCTION AND APPLICATION SCHEMATICS.....	9
3.3 MEASUREMENT OF PRESSURE AND TEMPERATURE	11
3.3.1 Hardware pressure sampling accuracy modes	12
3.3.2 Software pressure sampling accuracy modes	13
3.4 CALIBRATION COEFFICIENTS.....	13
3.5 CALCULATING PRESSURE AND TEMPERATURE	14
3.6 CALCULATING ABSOLUTE ALTITUDE.....	16
3.7 CALCULATING PRESSURE AT SEA LEVEL	17
4. GLOBAL MEMORY MAP	18
5. I²C INTERFACE	19
5.1 I ² C SPECIFICATION.....	19
5.2 DEVICE AND REGISTER ADDRESS.....	20
5.3 I ² C PROTOCOL	20
5.4 START TEMPERATURE AND PRESSURE MEASUREMENT	21
5.5 READ A/D CONVERSION RESULT OR E ² PROM DATA	22
6. PACKAGE.....	23
6.1 PIN CONFIGURATION	23
6.2 OUTLINE DIMENSIONS	24
6.2.1 Bottom view	24
6.2.2 Top view	25
6.2.3 Side view	25
6.3 MOISTURE SENSITIVITY LEVEL AND SOLDERING.....	26
6.4 ROHS COMPLIANCY.....	26
6.5 MOUNTING AND ASSEMBLY RECOMMENDATIONS	26
7. LEGAL DISCLAIMER.....	27

BST-BMP180-DS000-09 | Revision 2.5 | April 2013

Bosch Sensortec


© Bosch Sensortec GmbH reserves all rights even in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. BOSCH and the symbol are registered trademarks of Robert Bosch GmbH, Germany.
Note: Specifications within this document are subject to change without notice.




 BOSCH	Data sheet BMP180	Page 5
--	-----------------------------	--------

7.1 ENGINEERING SAMPLES	27
7.2 PRODUCT USE.....	27
7.3 APPLICATION EXAMPLES AND HINTS.....	27
8. DOCUMENT HISTORY AND MODIFICATION.....	28



 BOSCH	Data sheet BMP180	Page 5
--	-----------------------------	--------

7.1 ENGINEERING SAMPLES	27
7.2 PRODUCT USE.....	27
7.3 APPLICATION EXAMPLES AND HINTS.....	27
8. DOCUMENT HISTORY AND MODIFICATION.....	28

 BOSCH	Data sheet BMP180	Page 6
--	-----------------------------	--------

1. Electrical characteristics

If not stated otherwise, the given values are ± 3 -Sigma values over temperature/voltage range in the given operation mode. All values represent the new parts specification; additional solder drift is shown separately.

Table 1: Operating conditions, output signal and mechanical characteristics


Parameter	Symbol	Condition	Min	Typ	Max	Units
Operating temperature	T_A	operational	-40		+85	$^{\circ}\text{C}$
		full accuracy	0		+65	
Supply voltage	V_{DD}	ripple max. 50mVpp	1.8	2.5	3.6	V
			1.62	2.5	3.6	
Supply current @ 1 sample / sec. 25 $^{\circ}\text{C}$	I_{DDLOW}	ultra low power mode		3		μA
	I_{DDSTD}	standard mode		5		μA
	I_{DDHR}	high resolution mode		7		μA
	I_{DDUHR}	Ultra high res. mode		12		μA
	I_{DDAR}	Advanced res. mode		32		μA
Peak current	I_{peak}	during conversion		650	1000	μA
Standby current	I_{DDBM}	@ 25 $^{\circ}\text{C}$		0.1	4 ¹	μA
Relative accuracy pressure $V_{DD} = 3.3\text{V}$		950 ... 1050 hPa @ 25 $^{\circ}\text{C}$		± 0.12		hPa
				± 1.0		m
		700 ... 900hPa 25 ... 40 $^{\circ}\text{C}$		± 0.12		hPa
Absolute accuracy pressure $V_{DD} = 3.3\text{V}$				± 1.0		m
		300 ... 1100 hPa 0 ... +65 $^{\circ}\text{C}$	-4.0	-1.0*	+2.0	hPa
		300 ... 1100 hPa -20 ... 0 $^{\circ}\text{C}$	-6.0	-1.0*	+4.5	hPa
Resolution of output data		pressure		0.01		hPa
		temperature		0.1		$^{\circ}\text{C}$
Noise in pressure		see table on page 12-13				
Absolute accuracy temperature $V_{DD} = 3.3\text{V}$		@ 25 $^{\circ}\text{C}$	-1.5	± 0.5	+1.5	$^{\circ}\text{C}$
		0 ... +65 $^{\circ}\text{C}$	-2.0	± 1.0	+2.0	$^{\circ}\text{C}$

¹ at 85 $^{\circ}\text{C}$

BST-BMP180-DS000-09 | Revision 2.5 | April 2013

Bosch Sensortec


© Bosch Sensortec GmbH reserves all rights even in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. BOSCH and the symbol are registered trademarks of Robert Bosch GmbH, Germany.
Note: Specifications within this document are subject to change without notice.

 BOSCH	Data sheet BMP180	Page 7
--	-----------------------------	--------

Conversion time pressure	$t_{c_p_low}$	ultra low power mode		3	4.5	ms
	$t_{c_p_std}$	standard mode		5	7.5	ms
	$t_{c_p_hr}$	high resolution mode		9	13.5	ms
	$t_{c_p_luhr}$	ultra high res. mode		17	25.5	ms
	$t_{c_p_ar}$	Advanced res. mode		51	76.5	ms
Conversion time temperature	t_{c_temp}	standard mode		3	4.5	ms
Serial data clock	f_{SCL}				3.4	MHz
Solder drifts		Minimum solder height 50µm	-0.5		+2	hPa
Long term stability**		12 months		±1.0		hPa

* The typical value is: -1±1

** Long term stability is specified in the full accuracy operating pressure range 0 ... 65°C

 BOSCH	Data sheet BMP180	Page 8
--	-----------------------------	--------

2. Absolute maximum ratings

Table 2: Absolute maximum ratings


Parameter	Condition	Min	Max	Units
Storage temperature		-40	+85	°C
Supply voltage	all pins	-0.3	+4.25	V
ESD rating	HBM, R = 1.5kΩ, C = 100pF		±2	kV
Overpressure			10,000	hPa

The BMP180 has to be handled as

Electrostatic Sensitive Device (ESD).



Figure 1: ESD

 BOSCH	Data sheet BMP180	Page 9
--	-----------------------------	--------

3. Operation

3.1 General description

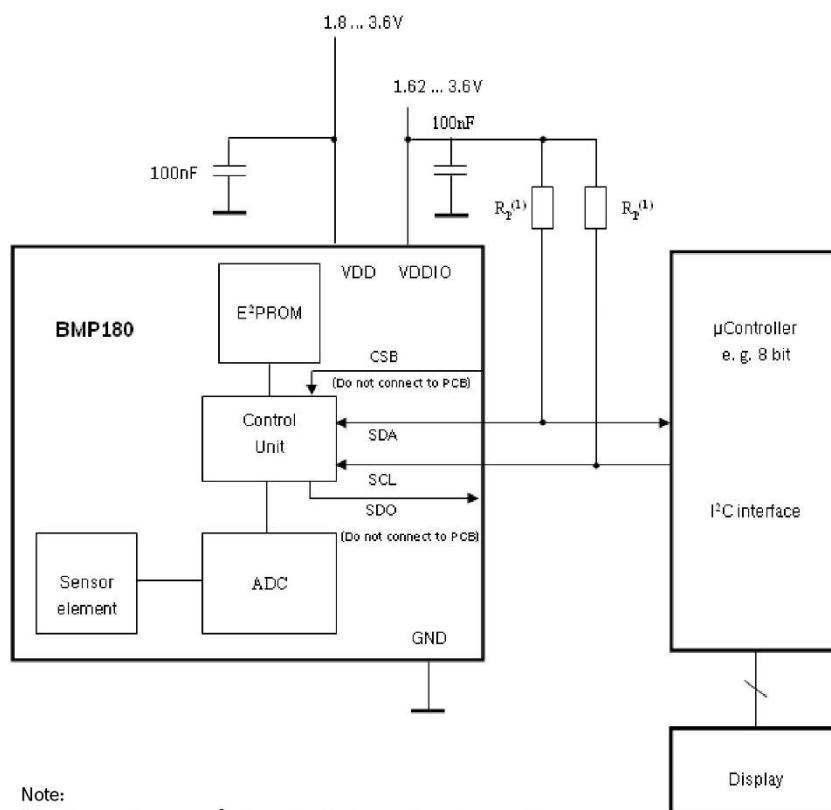
The BMP180 is designed to be connected directly to a microcontroller of a mobile device via the I²C bus. The pressure and temperature data has to be compensated by the calibration data of the E²PROM of the BMP180.

3.2 General function and application schematics

The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with E²PROM and a serial I²C interface. The BMP180 delivers the uncompensated value of pressure and temperature. The E²PROM has stored 176 bit of individual calibration data. This is used to compensate offset, temperature dependence and other parameters of the sensor.

- UP = pressure data (16 to 19 bit)
- UT = temperature data (16 bit)


BOSCH	Data sheet BMP180	Page 10
--------------	-----------------------------	---------



Note:

(1) Pull-up resistors for I²C bus, R_p = 2.2kΩ ... 10kΩ, typ. 4.7kΩ

Figure 2: Typical application circuit

 BOSCH	Data sheet BMP180	Page 11
--	-----------------------------	---------

3.3 Measurement of pressure and temperature

For all calculations presented here an ANSI C code is available from Bosch Sensortec ("BMP180 _API").

The microcontroller sends a start sequence to start a pressure or temperature measurement. After converting time, the result value (UP or UT, respectively) can be read via the I²C interface. For calculating temperature in °C and pressure in hPa, the calibration data has to be used. These constants can be read out from the BMP180 E²PROM via the I²C interface at software initialization.

The sampling rate can be increased up to 128 samples per second (standard mode) for dynamic measurement. In this case, it is sufficient to measure the temperature only once per second and to use this value for all pressure measurements during the same period.

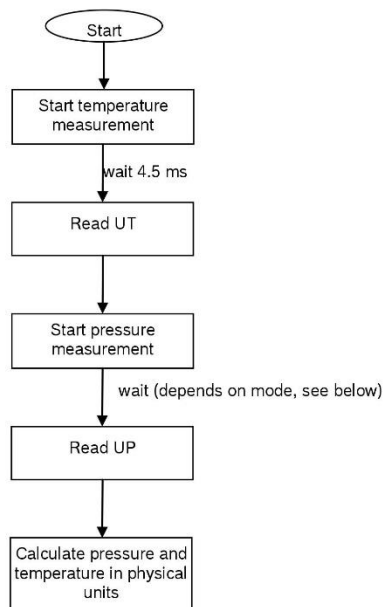



Figure 3: Measurement flow BMP180

 BOSCH	Data sheet BMP180	Page 12
--	-----------------------------	---------

3.3.1 Hardware pressure sampling accuracy modes


By using different modes the optimum compromise between power consumption, speed and resolution can be selected, see below table.

Table 3: Overview of BMP180 hardware accuracy modes, selected by driver software via the variable *oversampling_setting*

Mode	Parameter <i>oversampling_setting</i>	Internal number of samples	Conversion time pressure max. [ms]	Avg. current @ 1 sample/s typ. [μA]	RMS noise typ. [hPa]	RMS noise typ. [m]
ultra low power	0	1	4.5	3	0.06	0.5
standard	1	2	7.5	5	0.05	0.4
high resolution	2	4	13.5	7	0.04	0.3
ultra high resolution	3	8	25.5	12	0.03	0.25

For further information on noise characteristics see the relevant application note "Noise in pressure sensor applications".

All modes can be performed at higher speeds, e.g. up to 128 times per second for standard mode, with the current consumption increasing proportionally to the sample rate.

 BOSCH	Data sheet BMP180	Page 13
--	-----------------------------	---------

3.3.2 Software pressure sampling accuracy modes

For applications where a low noise level is critical, averaging is recommended if the lower bandwidth is acceptable. Oversampling can be enabled using the software API driver (with OSR = 3).

Table 4: Overview of BMP180 software accuracy mode, selected by driver software via the variable *software_oversampling_setting*

Mode	Parameter <i>oversampling_setting</i>	software_ oversampl ing_settin g	Conversion time pressure max. [ms]	Avg. current @ 1 sample/s typ. [μA]	RMS noise typ. [hPa]	RMS noise typ. [m]
Advanced resolution	3	1	76.5	32	0.02	0.17


3.4 Calibration coefficients

The 176 bit E²PROM is partitioned in 11 words of 16 bit each. These contain 11 calibration coefficients. Every sensor module has individual coefficients. Before the first calculation of temperature and pressure, the master reads out the E²PROM data.

The data communication can be checked by checking that none of the words has the value 0 or 0xFFFF.

Table 5: Calibration coefficients

Parameter	BMP180 reg adr	
	MSB	LSB
AC1	0xAA	0xAB
AC2	0xAC	0xAD
AC3	0xAE	0xAF
AC4	0xB0	0xB1
AC5	0xB2	0xB3
AC6	0xB4	0xB5
B1	0xB6	0xB7
B2	0xB8	0xB9
MB	0xBA	0xBB
MC	0xBC	0xBD
MD	0xBE	0xBF

 BOSCH	Data sheet BMP180	Page 14
--	-----------------------------	---------

3.5 Calculating pressure and temperature

The mode (ultra low power, standard, high, ultra high resolution) can be selected by the variable *oversampling_setting* (0, 1, 2, 3) in the C code.

Calculation of true temperature and pressure in steps of 1Pa (= 0.01hPa = 0.01mbar) and temperature in steps of 0.1°C.

The following figure shows the detailed algorithm for pressure and temperature measurement.

This algorithm is available to customers as reference C source code ("BMP180_API") from Bosch Sensortec and via its sales and distribution partners. **Please contact your Bosch Sensortec representative for details.**

BOSCH	Data sheet BMP180	Page 15
--------------	------------------------------------	---------

Calculation of pressure and temperature for BMP180

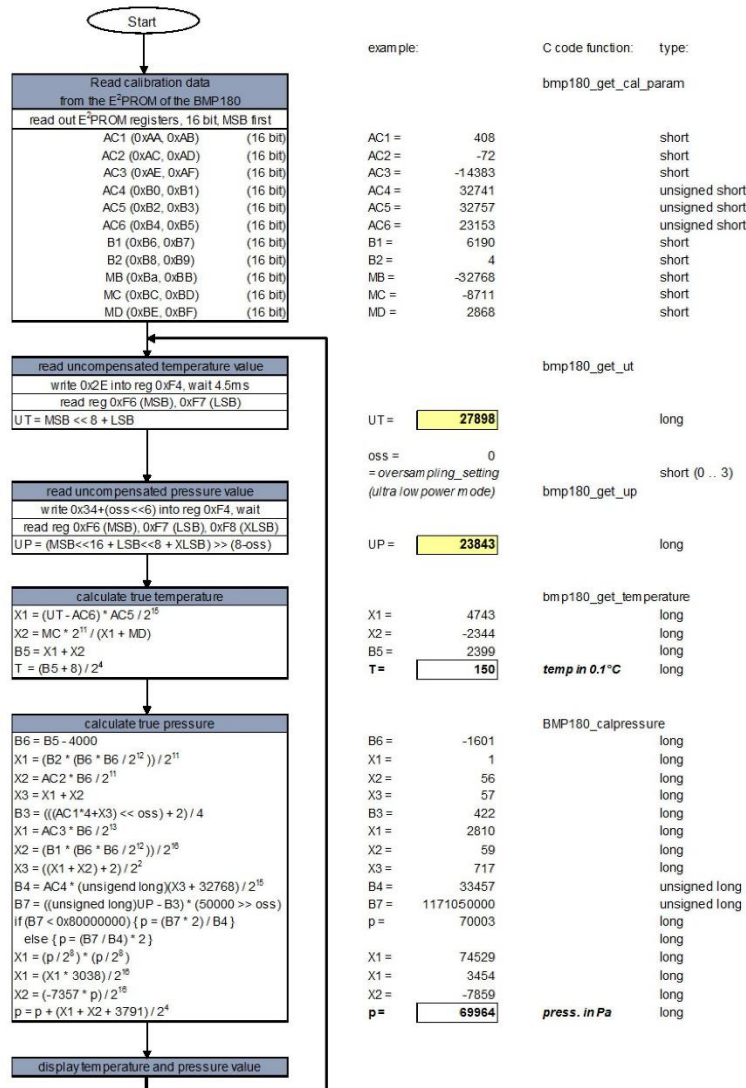



Figure 4: Algorithm for pressure and temperature measurement

 BOSCH	<p>Data sheet BMP180</p>	<p>Page 16</p>
--	------------------------------	----------------

3.6 Calculating absolute altitude

With the measured pressure p and the pressure at sea level p_0 e.g. 1013.25hPa, the altitude in meters can be calculated with the international barometric formula:

$$\text{altitude} = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right)$$

Thus, a pressure change of $\Delta p = 1\text{hPa}$ corresponds to 8.43m at sea level.

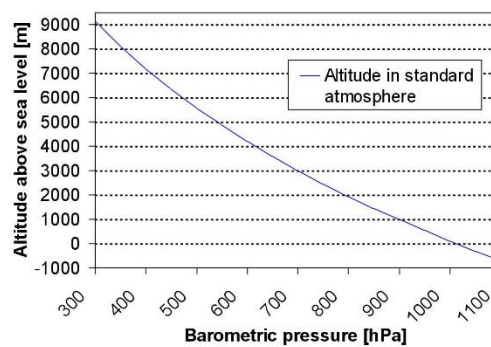



Figure 5: Transfer function: Altitude over sea level – Barometric pressure


 BOSCH	<p>Data sheet BMP180</p>	<p>Page 17</p>
--	-------------------------------------	----------------

3.7 Calculating pressure at sea level

With the measured pressure p and the absolute altitude the pressure at sea level can be calculated:

$$p_0 = \frac{p}{\left(1 - \frac{\text{altitude}}{44330}\right)^{5.255}}$$

Thus, a difference in altitude of $\Delta\text{altitude} = 10\text{m}$ corresponds to 1.2hPa pressure change at sea level.

 BOSCH	Data sheet BMP180	Page 18
--	-----------------------------	---------

4. Global Memory Map

The memory map below shows all externally accessible data registers which are needed to operate BMP180. The left columns show the memory addresses. The columns in the middle depict the content of each register bit. The colors of the bits indicate whether they are read-only, write-only or read- and writable. The memory is volatile so that the writable content has to be re-written after each power-on.

Not all register addresses are shown. These registers are reserved for further Bosch factory testing and trimming.

Register Name	Register Address	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Reset state
out_xlsb	F8h	adc_out_xlsb<7:3>					0	0	0	00h
out_lsb	F7h	adc_out_lsb<7:0>								00h
out_msb	F6h	adc_out_msb<7:0>								80h
ctrl_meas	F4h	oss<1:0>		sco	measurement control					00h
soft reset	E0h						reset			00h
id	D0h						id<7:0>			55h
calib21 downto calib0	BFh downto AAh	calib21<7:0> downto calib0<7:0>								n/a

Registers:

Control registers	Calibration registers	Data registers	Fixed
Type: read / write	read only	read only	read only

Figure 6: Memory map

Measurement control (register F4h <4:0>): Controls measurements. Refer to Figure 6 for usage details.


Sco (register F4h <5>): Start of conversion. The value of this bit stays "1" during conversion and is reset to "0" after conversion is complete (data registers are filled).

Oss (register F4h <7:6>): controls the oversampling ratio of the pressure measurement (00b: single, 01b: 2 times, 10b: 4 times, 11b: 8 times).

Soft reset (register E0h): Write only register. If set to 0xB6, will perform the same sequence as power on reset.

Chip-id (register D0h): This value is fixed to 0x55 and can be used to check whether communication is functioning.

After conversion, data registers can be read out in any sequence (i.e. MSB first or LSB first). Using a burst read is not mandatory.

 BOSCH	Data sheet BMP180	Page 19
--	-----------------------------	---------

5. I²C Interface

- I²C is a digital two wire interface
- Clock frequencies up to 3.4Mbit/sec. (I²C standard, fast and high-speed mode supported)
- SCL and SDA needs a pull-up resistor, typ. 4.7kOhm to V_{DDIO}
(one resistor each for all the I²C bus)

The I²C bus is used to control the sensor, to read calibration data from the E²PROM and to read the measurement data when A/D conversion is finished. SDA (serial data) and SCL (serial clock) have open-drain outputs.


For detailed I²C-bus specification please refer to:

http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf

5.1 I²C specification

Table 6: Electrical parameters for the I²C interface

Parameter	Symbol	Min.	Typ	Max.	Units
Clock input frequency	f _{SCL}			3.4	MHz
Input-low level	V _{IL}	0		0.2 * V _{DDIO}	V
Input-high level	V _{IH}	0.8 * V _{DDIO}		V _{DDIO}	V
Voltage output low level @ V _{DDIO} = 1.62V, I _{OL} = 3mA	V _{OL}			0.3	V
SDA and SCL pull-up resistor	R _{pull-up}	2.2		10	kOhm
SDA sink current @ V _{DDIO} = 1.62V, V _{OL} = 0.3V	I _{SDA_sink}		9		mA
Start-up time after power-up, before first communication	t _{Start}	10			Ms

 BOSCH	Data sheet BMP180	Page 20
--	-----------------------------	---------

5.2 Device and register address

The BMP180 module address is shown below. The LSB of the device address distinguishes between read (1) and write (0) operation, corresponding to address 0xEF (read) and 0xEE (write).

Table 7: BMP180 addresses

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	1	0	1	1	1	0/1

5.3 I²C protocol

The I²C interface protocol has special bus signal conditions. Start (S), stop (P) and binary data conditions are shown below. At start condition, SCL is high and SDA has a falling edge. Then the slave address is sent. After the 7 address bits, the direction control bit R/W selects the read or write operation. When a slave device recognizes that it is being addressed, it should acknowledge by pulling SDA low in the ninth SCL (ACK) cycle.

At stop condition, SCL is also high, but SDA has a rising edge. Data must be held stable at SDA when SCL is high. Data can change value at SDA only when SCL is low.

Even though V_{DDIO} can be powered on before V_{DD}, there is a chance of excessive power consumption (a few mA) if this sequence is used, and the state of the output pins is undefined so that the bus can be locked. Therefore, V_{DD} *must* be powered before V_{DDIO} unless the limitations above are understood and not critical.

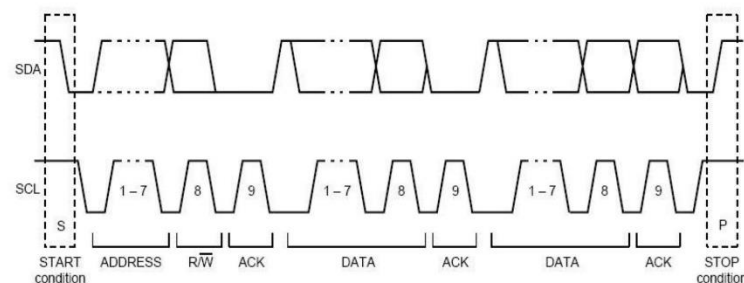



Figure 7: I²C protocol

 BOSCH	Data sheet BMP180	Page 21
--	-----------------------------	---------

5.4 Start temperature and pressure measurement

The timing diagrams to start the measurement of the temperature value UT and pressure value UP are shown below. After start condition the master sends the device address write, the register address and the control register data. The BMP180 sends an acknowledgement (ACKS) every 8 data bits when data is received. The master sends a stop condition after the last ACKS.

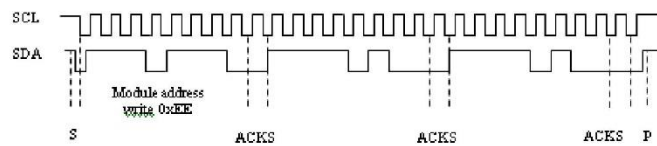



Figure 8: Timing diagram for starting pressure measurement

Abbreviations:

S	Start
P	Stop
ACKS	Acknowledge by Slave
ACKM	Acknowledge by Master
NACKM	Not Acknowledge by Master

Table 8: Control registers values for different internal oversampling_setting (oss)

Measurement	Control register value (register address 0xF4)	Max. conversion time [ms]
Temperature	0x2E	4.5
Pressure (oss = 0)	0x34	4.5
Pressure (oss = 1)	0x74	7.5
Pressure (oss = 2)	0xB4	13.5
Pressure (oss = 3)	0xF4	25.5

 BOSCH	Data sheet BMP180	Page 22
--	-----------------------------	---------

5.5 Read A/D conversion result or E²PROM data

To read out the temperature data word UT (16 bit), the pressure data word UP (16 to 19 bit) and the E²PROM data proceed as follows:

After the start condition the master sends the module address write command and register address. The register address selects the read register:

E²PROM data registers 0xAA to 0xBF

Temperature or pressure value UT or UP 0xF6 (MSB), 0xF7 (LSB), optionally 0xF8 (XLSB)

Then the master sends a restart condition followed by the module address read that will be acknowledged by the BMP180 (ACKS). The BMP180 sends first the 8 MSB, acknowledged by the master (ACKM), then the 8 LSB. The master sends a "not acknowledge" (NACKM) and finally a stop condition.

Optionally for ultra high resolution, the XLSB register with address 0xF8 can be read to extend the 16 bit word to up to 19 bits; refer to the application programming interface (API) software rev. 1.1 ("BMP180_API", available from Bosch Sensortec).

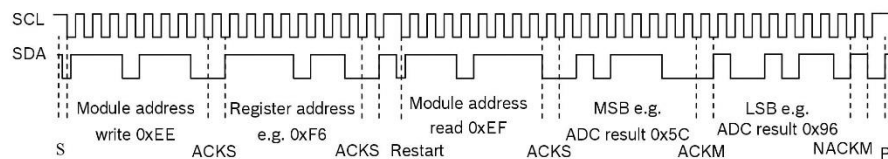



Figure 9: Timing diagram read 16 bit A/D conversion result

 BOSCH	Data sheet BMP180	Page 23
--	-----------------------------	---------

6. Package

6.1 Pin configuration

Picture shows the device in top view. Device pins are shown here transparently only for orientation purposes.

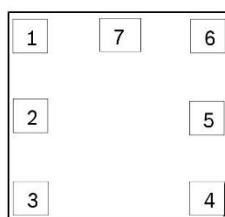



Figure 10: Layout pin configuration BMP180

Table 9: Pin configuration BMP180

in No	Name	Function
1	CSB*	Chip select
2	VDD	Power supply
3	VDDIO	Digital power supply
4	SDO*	SPI output
5	SCL	I2C serial bus clock input
6	SDA	I2C serial bus data (or SPI input)
7	GND	Ground

* A pin compatible product variant with SPI interface is possible upon customer's request. For I²C (standard case) CSB and SDO are not used, they have to be left open.
 All pins have to be soldered to the PCB for symmetrical stress input even though they are not connected internally.

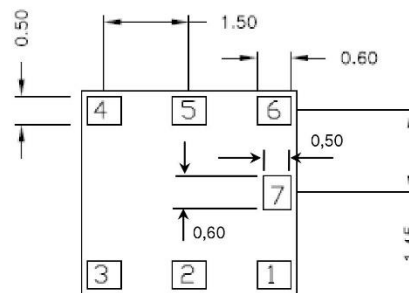
	<p>Data sheet BMP180</p>	<p>Page 24</p>
---	-------------------------------------	----------------

6.2 Outline dimensions

The sensor housing is a 7Pin LGA package with metal lid. Its dimensions are 3.60mm (± 0.1 mm) x 3.80mm (± 0.1 mm) x 0.93mm (± 0.07 mm).

Note: All dimensions are in mm.

6.2.1 Bottom view



BOTTOM VIEW

Figure 11: Bottom view BMP180

Orthographic projection of a mechanical part showing the top view. The part is a square plate with a central circular hole. Dimensions are given in millimeters.

Overall width: $3.60^{+0.10}_{-0.10}$

Overall height: $3.60^{+0.10}_{-0.10}$

Distance from top edge to center of hole: 1.225

Distance from left edge to center of hole: 3.35

Distance from bottom edge to center of hole: 2.85

Hole diameter: $\varnothing 0.50$


Bottom edge fillet radius: 0.850

Top view label: TOP VIEW

SIDE VIEW

BST-BMP180-DS000-09 | Revision 2.5 | April 2013

© Bosch Sensortec GmbH reserves all rights even in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. BOSCH and the symbol are registered trademarks of Robert Bosch GmbH, Germany.
Note: Specifications within this document are subject to change without notice.

 BOSCH	Data sheet BMP180	Page 26
--	-----------------------------	---------

6.3 Moisture sensitivity level and soldering

The BMP180 is classified MSL 1 (moisture sensitivity level) according to IPC/JEDEC standards J-STD-020D and J-STD-033A.

The device can be soldered Pb-free with a peak temperature of 260°C for 20 to 40 sec. The minimum height of the solder after reflow shall be at least 50µm. This is required for good mechanical decoupling between the sensor device and the printed circuit board (PCB).

To ensure good solder-ability, the devices shall be stored at room temperature (20°C).

The soldering process can lead to an offset shift.

6.4 RoHS compliancy

The BMP180 sensor meets the requirements of the EC directive "Restriction of hazardous substances (RoHS)", please refer also to:

"Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment".

The BMP180 sensor is also halogen-free.

6.5 Mounting and assembly recommendations


In order to achieve the specified performance for you design, the following recommendations and the "Handling, soldering & mounting instructions BMP180" should be taken into consideration when mounting a pressure sensor on a printed-circuit board (PCB):

- The clearance above the metal lid shall be 0.1mm at minimum.
- For the device housing appropriate venting needs to be provided in case the ambient pressure shall be measured.
- Liquids shall not come into direct contact with the device.
- During operation the sensor is sensitive to light, which can influence the accuracy of the measurement (photo-current of silicon).
- The BMP180 shall not be placed close the fast heating parts. In case of gradients > 3°C/sec. it is recommended to follow Bosch Sensortec application note ANP015, "Correction of errors induced by fast temperature changes". Please contact your Bosch Sensortec representative for details.

BST-BMP180-DS000-09 | Revision 2.5 | April 2013

Bosch Sensortec

© Bosch Sensortec GmbH reserves all rights even in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. BOSCH and the symbol are registered trademarks of Robert Bosch GmbH, Germany.
Note: Specifications within this document are subject to change without notice.

 BOSCH	Data sheet BMP180	Page 27
--	-----------------------------	---------

7. Legal disclaimer

7.1 Engineering samples

Engineering Samples are marked with an asterisk (*) or (e). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

7.2 Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or security sensitive systems. Security sensitive systems are those for which a malfunction is expected to lead to bodily harm or significant property damage. In addition, they are not fit for use in products which interact with motor vehicle systems.


The resale and/or use of products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the Purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser must monitor the market for the purchased products, particularly with regard to product safety, and inform Bosch Sensortec without delay of all security relevant incidents.

7.3 Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

 BOSCH	Data sheet BMP180	Page 28
--	-----------------------------	---------

8. Document history and modification

Rev. No	Chapter	Description of modifications/changes	Date
1.0		First edition for description of serial production material – Preliminary version	
1.1	5.1	New nomenclature of pin configuration	27 July 2010
1.2	5	Design change in package – hole in Lid and without slit	13 September 2010
1.3	3.2 5.1	- Standardizing pin naming over Bosch Sensortec products – typical application circuit - Optimizing pin description, SPI description	15 December 2010
2.0	1	- Non-preliminary version - Verifying parameter through characterization	28 January 2011
2.1	3.2 4 5.3 6.1 6.2.1	- Declaration of SDO and CSB pins in the typical application circuit - Adding global memory map and bits description - Power-up sequence - Description of used interfaces - Dimension pin7	1 April 2011
2.2	6.1	Correction of the pin configuration (editorial change)	14 April 2011
2.3	3.3	Optimizing noise performance	25 May 2011
2.4	6.3 page 2 5.1 5.3	Removed shelf-life constraints Comparison removed Voltage output low level added Power on sequence of V_{DD} and V_{DDIO} defined	26 January 2012
2.5	1 1 Figure 4 Page 2	Added max values for supply current for restricted version Added max value for standby current for restricted version Update of calculation of algorithm for pressure and temperature measurement Changed wording from “ultra high resolution mode” to “advanced resolution mode”	15 Feb 2013 5 Apr 2013

Bosch Sensortec GmbH
Gerhard-Kindler-Strasse 8
72770 Reutlingen / Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved | Printed in Germany
Specifications are subject to change without notice
Revision_2.5_042013
Document number: BST-BMP180-DS000-09

BST-BMP180-DS000-09 | Revision 2.5 | April 2013

Bosch Sensortec

© Bosch Sensortec GmbH reserves all rights even in the event of industrial property rights. We reserve all rights of disposal such as copying and passing on to third parties. BOSCH and the symbol are registered trademarks of Robert Bosch GmbH, Germany.
Note: Specifications within this document are subject to change without notice.

4. Mòdul bluetooth HC06 – DATASHEET

[16]

Guangzhou HC Information Technology Co., Ltd.

Guangzhou HC Information Technology Co., Ltd.

Product Data Sheet

Module Data Sheet

Rev 1

1. 0	2.0	2.1	2.2				
2006/6/18	2006/9/6	2010/4/22	2011/4/6				

DRAWN BY :	Ling Xin		MODEL : HC-06	
CHECKED BY :	Eric Huang		Description:: BC04 has external 8M Flash and EDR module HC-06 is industrial, and compatible with civil HC-04	
APPD. BY:	Simon Mok		REV: 2.0	Page :
Former version introduction	HC-06 is the higher version of LV_BC_2.0. Linvor is the former of wavesen.			

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com



Contents

1. Product's picture
2. Feature
3. Pins description
4. The parameters and mode of product
5. Block diagram
6. Debugging device
7. Characteristic of test
8. Test diagram
9. AT command set

Guangzhou HC Information Technology Co., Ltd.

1. Product's picture

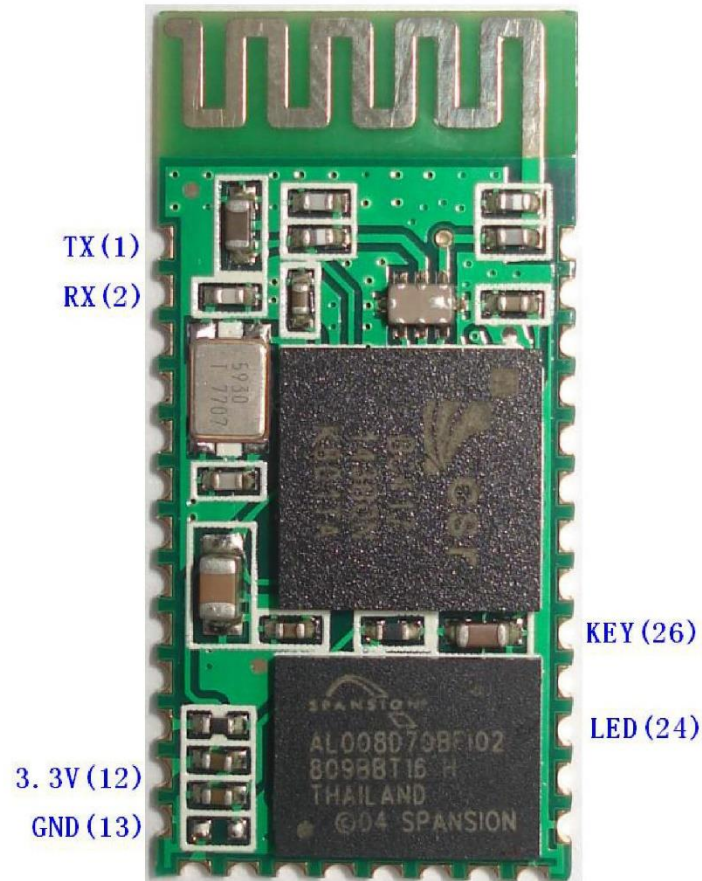


Figure 1 A Bluetooth module

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

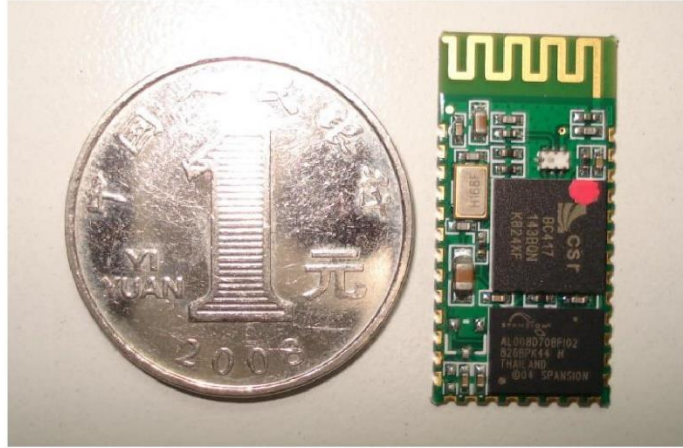


Figure 2. A Bluetooth module size



Figure 3 50 pieces chips in an anti-static blister package.

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

2. Feature

- Wireless transceiver
 - Sensitivity (Bit error rate) can reach -80dBm.
 - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
 - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
 - Has a build-in 2.4GHz antenna; user needn't test antenna.
 - Has the external 8Mbit FLASH
 - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA.
The current in communication is 8mA.
 - Standard HCI Port (UART or USB)
 - USB Protocol: Full Speed USB1.1, Compliant With 2.0
 - This module can be used in the SMD.
 - It's made through RoHS process.
 - The board PIN is half hole size.
 - Has a 2.4GHz digital wireless transceiver.
 - Bases at CSR BC04 Bluetooth technology.
 - Has the function of adaptive frequency hopping.
 - Small (27mm×13mm×2mm)
 - Peripherals circuit is simple.
 - It's at the Bluetooth class 2 power level.
 - Storage temperature range: -40 °C - 85°C, work temperature range: -25 °C - +75°C
 - Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
 - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.
- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

- Application fields:
 - Bluetooth Car Handsfree Device
 - Bluetooth GPS
 - Bluetooth PCMCIA , USB Dongle
 - Bluetooth Data Transfer
- Software
 - CSR

3. PINs description

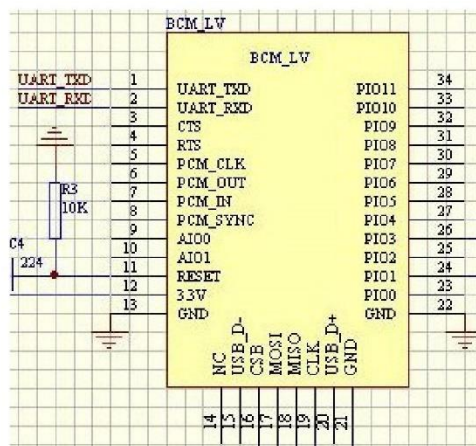


Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
VCC	12	3.3V		
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA(if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA(if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	CLK_REQ
PIO7	30	Bi-Directional	Programmable input/output line	CLK_OUT
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	
RESETB	11	CMOS Input with weak internal pull-down		
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal	Chip select for serial peripheral interface, active low	

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

		pull-up		
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_-	15	Bi-Directional		
USB_+	20	Bi-Directional		
1.8V	14		1.8V external power supply input	Default : 1.8V internal power supply.
PCM_CLK	5	Bi-Directional		
PCM_OUT	6	CMOS output		
PCM_IN	7	CMOS Input		
PCM_SYNC	8	Bi-Directional		

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com



4. The parameters and mode of product

LINVOR BLUE T

www.linvor.com

Bluetooth Module



CSR,BC417143B

V 2.0

2006/09/6

蓝牙 RF 模块

1. 采用 CSR BC4 +8M FLASH 方案
2. 具有 PIO0-PIO11、AIO0、AIO1、
USB、PCM、UART 及 SPI 接口，
模块内置 8MFLASH，功能强大，
用户可定制软件,适用于各种蓝牙
设备，内置 RF 天线,便于调试。

蓝牙协议版本	Bluetooth Specification V2.0 With EDR
USB 协议	Full Speed USB V1.1
USB Protocol	Compliant With USB V2.0
频率	2.4Ghz ISM band
调制方式	GFSK(Gaussian Frequency Shift Keying)
发射功率	-4 ~+4 dBm, Class 2
灵敏度	≤-80dBm at 0.1% BER
通讯速率	Asynchronous:2Mbps(Max)
供电电源	3.3V
工作温度	-20~+55 Centigrade
封装尺寸	27mmX13mmX2mm

Page 1 of 2

If you want more information, please visit www.wavesen.com.

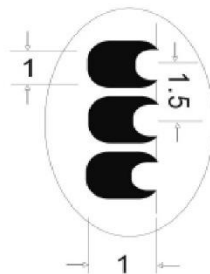
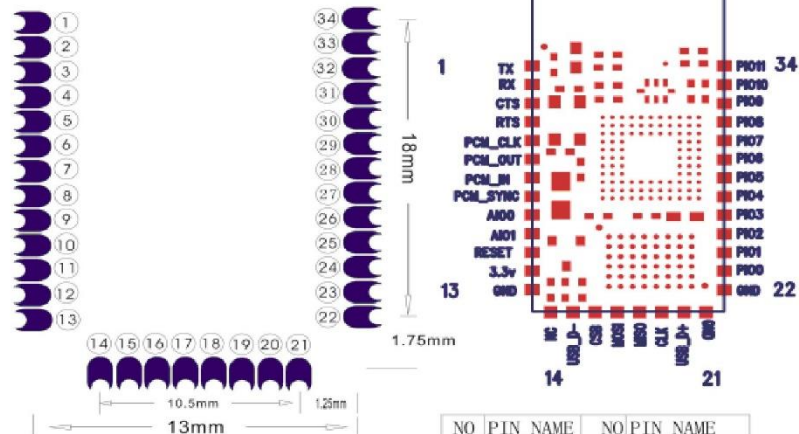
www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

LINVOR BLUE T
 www.linvor.com

LV-BC-2.0

单位：mm



PCB Layout 请参考实物

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

5. Block diagram

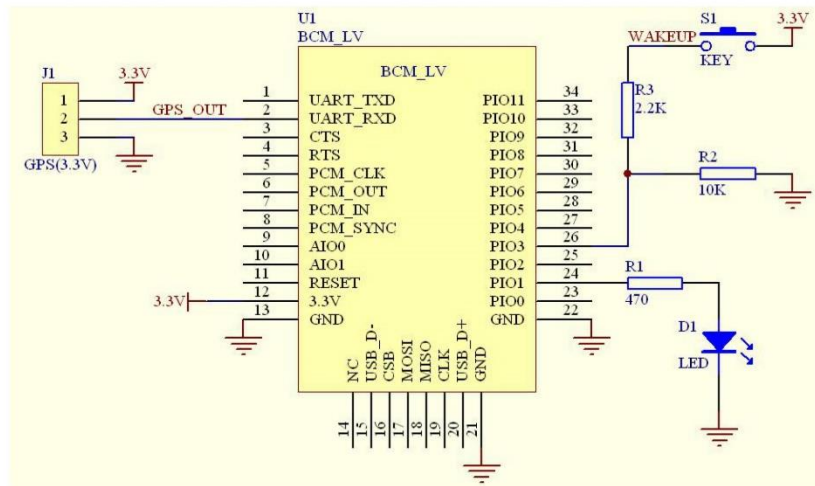


Figure 5 Block diagram 1

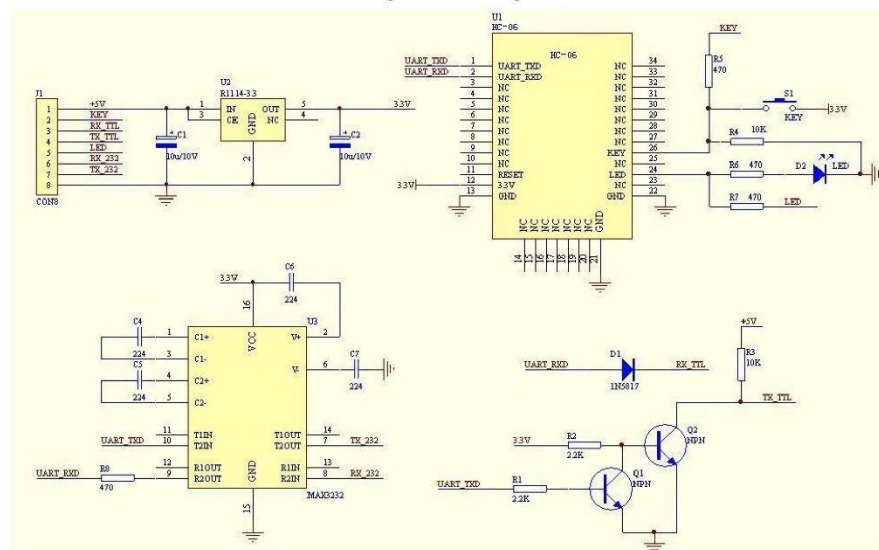


Figure 5 Block diagram 2

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

HC-04/06 master device has a function of remembering the last paired slave device. As a master device, it will search the last paired slave device until the connection is built. But if the WAKEUP button is pressed, HC-04/06 will lose the memory and research the new slave device.

6. Debugging device

6.1 Device

PC, hardware, 3G, 3G Frequency Counter (SP3386), 3.15V DC power supply, Shielding, Bluetooth

Test box.

6.2 Software

7. Characteristic of test

Test Condition 25°C RH 65%				
	Min	Typ	Max	Unit
1. Carrier Freq. (ISM Band)	2.4		2.4835	MHz
2. RF O/P Power	-6	2	4	dBm
3. Step size of Power control	2		8	dB
4. Freq. Offset (Typical Carrier freq.)	-75		75	KHz
5. Carrier Freq. drift (Hopping on, drift rate/50uS)	-20		20	KHz
1 slot packet	-25		25	KHz
3 slot packet	-40		-40	KHz
6. Average Freq. Deviations (Hopping off, modulation)	140		175	KHz
Freq. Deviation	115			KHz
Ratio of Freq. Deviation	0.8			
7. Receive Sensitivity @< 0.1% BER (Bit error rate)	-83			dBm

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

8. Test diagram

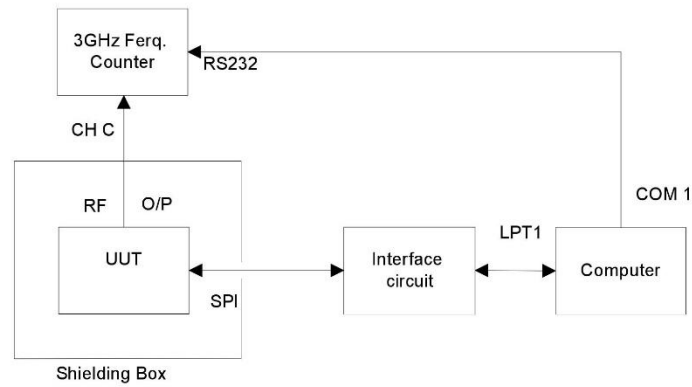


Fig 1. Programming and Freq. Alignment

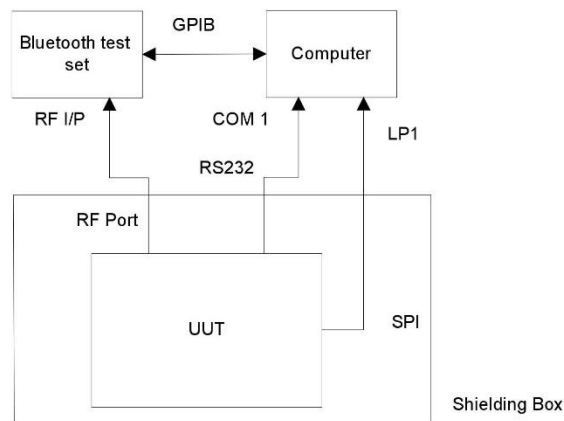


Fig 2 RF parameter Test Procedure

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

Guangzhou HC Information Technology Co., Ltd.

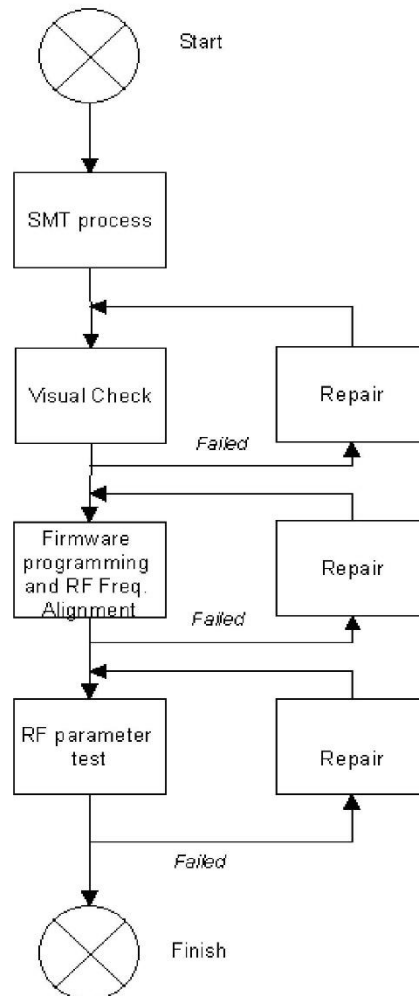


Fig 3 Assemble/Alignment/Testing Flow Chart

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

9. AT command set

The way to the AT command mode: supply power to the module, it will enter to the AT mode if it needn't pair. The interval of command is about 1 second.

Default parameter: Baud rate:9600N81, ID: linvor, Password:1234

1. Test communication

Send: AT (please send it every second)

Back: OK

2. Reset the Bluetooth serial baud rate

Send: AT+BAUD1

Back: OK1200

Send: AT+BAUD2

Back: OK2400

.....

1-----1200

2-----2400

3-----4800

4-----9600 (Default)

5-----19200

6-----38400

7-----57600

8-----115200

9-----230400

A-----460800

B-----921600

C-----1382400

PC can't support the baud rate larger than 115200. The solution is: make the MCU have higher baud rate (larger than 115200) through programming, and reset the baud rate to low level through the AT command.

The baud rate reset by the AT command can be kept for the next time even though the power is cut off.

3. Reset the Bluetooth name

Send: AT+NAMEname

Back: OKname

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com



Guangzhou HC Information Technology Co., Ltd.

Parameter name: Name needed to be set (20 characters limited)

Example:

Send: AT+NAMEbill_gates

Back: OKname

Now, the Bluetooth name is reset to be "bill_gates"

The parameter can be kept even though the power is cut off. User can see the new Bluetooth name in PDA refresh service. (Note: The name is limited in 20 characters.)

4. change the Bluetooth pair password

Send: AT+PINxxxx

Back: OKsetpin

Parameter xxxx: The pair password needed to be set, is a 4-bits number. This command can be used in the master and slave module. At some occasions, the master module may be asked to enter the password when the master module tries to connect the slave module (adapter or cell-phone). Only if the password is entered, the successful connection can be built. At the other occasions, the pair can be finish automatically if the master module can search the proper slave module and the password is correct.

Besides the paired slave module, the master can connect the other devices who have slave module, such as Bluetooth digital camera, Bluetooth GPS, Bluetooth serial printer etc.

Example:

Send: AT+PIN8888

Back: OKsetpin

Then the password is changed to be 8888, while the default is 1234.

This parameter can be kept even though the power is cut off.

5. No parity check (The version, higher than V1.5, can use this command)

Send: AT+PN (This is the default value)

Back: OK NONE

6. Set odd parity check (The version, higher than V1.5, can use this command)

Send: AT+PO

Back: OK ODD

7. Set even parity check(The version, higher than V1.5, can use this command)

Send: AT+PE

Back: OK EVEN

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com



Guangzhou HC Information Technology Co., Ltd.

8. Get the AT version

Send: AT+VERSION

Back: LinvorV1.n

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
Complaint and suggestion: sunbirdit@hotmail.com

5. GPS NEO 6MV2 – u-blox – DATASHEET

[17]

GPS

NEO-6

u-blox 6 GPS Modules

Data Sheet

locate, communicate, accelerate

Abstract

Technical data sheet describing the cost effective, high-performance u-blox 6 based NEO-6 series of GPS modules, that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor.

These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements.



16.0 x 12.2 x 2.4 mm

www.u-blox.com



Document Information

Title	NEO-6
Subtitle	u-blox6 GPS Modules
Document type	Data Sheet
Document number	GPS.G6-HW-09005-E

Document status

Document status information

Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
NEO-6G	NEO-6G-0-001	ROM7.03	UBX-TN-11047-1
NEO-6Q	NEO-6Q-0-001	ROM7.03	UBX-TN-11047-1
NEO-6M	NEO-6M-0-001	ROM7.03	UBX-TN-11047-1
NEO-6P	NEO-6P-0-000	ROM6.02	N/A
NEO-6V	NEO-6V-0-000	ROM7.03	N/A
NEO-6T	NEO-6T-0-000	ROM7.03	N/A

This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.

u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2011, u-blox AG.

u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. ARM® is the registered trademark of ARM Limited in the EU and other countries.





Contents

Contents.....	3
1 Functional description.....	5
1.1 Overview	5
1.2 Product features	5
1.3 GPS performance.....	6
1.4 Block diagram.....	7
1.5 Assisted GPS (A-GPS).....	7
1.6 AssistNow Autonomous	7
1.7 Precision Timing.....	8
1.7.1 Time mode	8
1.7.2 Timepulse and frequency reference	8
1.7.3 Time mark	8
1.8 Raw data	8
1.9 Automotive Dead Reckoning	8
1.10 Precise Point Positioning.....	9
1.11 Oscillators	9
1.12 Protocols and interfaces	9
1.12.1 UART.....	9
1.12.2 USB	9
1.12.3 Serial Peripheral Interface (SPI).....	9
1.12.4 Display Data Channel (DDC)	10
1.13 Antenna.....	10
1.14 Power Management	10
1.14.1 Maximum Performance Mode	10
1.14.2 Eco Mode.....	10
1.14.3 Power Save Mode	11
1.15 Configuration	11
1.15.1 Boot-time configuration	11
1.16 Design-in	11
2 Pin Definition.....	12
2.1 Pin assignment	12
3 Electrical specifications	14
3.1 Absolute maximum ratings	14
3.2 Operating conditions	15
3.3 Indicative power requirements.....	15
3.4 SPI timing diagrams	16
3.4.1 Timing recommendations	16
4 Mechanical specifications	17



5	Qualification and certification.....	18
5.1	Reliability tests	18
5.2	Approvals	18
6	Product handling & soldering.....	19
6.1	Packaging	19
6.1.1	Reels	19
6.1.1	Tapes	20
6.2	Moisture Sensitivity Levels	21
6.3	Reflow soldering	21
6.4	ESD handling precautions	21
7	Default settings.....	22
8	Labeling and ordering information.....	23
8.1	Product labeling.....	23
8.2	Explanation of codes.....	23
8.3	Ordering information.....	24
	Related documents.....	24
	Revision history.....	24
	Contact.....	25



1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

1.2 Product features

Model	Type					Supply		Interfaces				Features						
	GPS	PPP	Timing	Raw Data	Dead Reckoning	1.75 V - 2.0 V	2.7 V - 3.6 V	UART	USB	SPi	I2C (I ² C compliant)	Programmable (flash) FW update	TCXO	RTC crystal	Antenna supply and supervisor	Configuration pins	Timepulse	External interrupt/ Wake-up
NEO-6G	●					●		●	●	●	●		●	●	○	3	1	●
NEO-6Q	●						●	●	●	●	●		●	●	○	3	1	●
NEO-6M	●						●	●	●	●	●			●	○	3	1	●
NEO-6P	●	●		●			●	●	●	●	●			●	○	3	1	●
NEO-6V	●				●		●	●	●	●	●			●	○	3	1	●
NEO-6T	●		●	●			●	●	●	●	●		●	●	○	3	1	●

○ = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series



All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.



1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix ¹		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Cold Start ²	26 s	27 s	32 s
	Warm Start ²	26 s	27 s	32 s
	Hot Start ²	1 s	1 s	1 s
	Aided Starts ³	1 s	<3 s	<3 s
Sensitivity ⁴		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition ⁵	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/M/T	NEO-6P/V	
		5Hz	1 Hz	
Horizontal position accuracy ⁶	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP ⁷	< 1 m (2D, R50) ⁸		
	SBAS + PPP ⁷	< 2 m (3D, R50) ⁸		
Configurable Timepulse frequency range		NEO-6G/Q/M/P/V	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated ⁹	15 ns		
Velocity accuracy ⁶		0.1 m/s		
Heading accuracy ⁶		0.5 degrees		
Operational Limits	Dynamics	< 4 g		
	Altitude ¹⁰	50,000 m		
	Velocity ¹⁰	500 m/s		

Table 2: NEO-6 GPS performance

¹ All satellites at -130 dBm

² Without aiding

³ Dependent on aiding data connection speed and latency

⁴ Demonstrated with a good active antenna

⁵ For an outage duration ≤10s

⁶ CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m

⁷ NEO-6P only

⁸ Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded. HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.

⁹ Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal

¹⁰ Assuming Airborne <4g platform



NEO-6 - Data Sheet

1.4 Block diagram

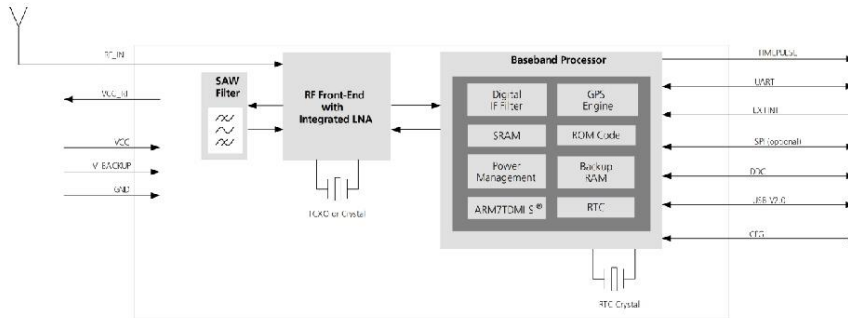


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

1.5 Assisted GPS (A-GPS)

Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status and an optional time synchronization signal will reduce time to first fix significantly and improve the acquisition sensitivity. All NEO-6 modules support the u-blox AssistNow Online and AssistNow Offline A-GPS services¹¹ and are OMA SUPL compliant.

1.6 AssistNow Autonomous

AssistNow Autonomous provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously broadcast satellite ephemeris data downloaded to and stored by the GPS receiver, AssistNow Autonomous automatically generates accurate satellite orbital data ("AssistNow Autonomous data") that is usable for future GPS position fixes. AssistNow Autonomous data is reliable for up to 3 days after initial capture.

u-blox' AssistNow Autonomous benefits are:

- Faster position fix
- No connectivity required
- Complementary with AssistNow Online and Offline services
- No integration effort, calculations are done in the background



For more details see the u-blox 6 Receiver Description including Protocol Specification [2].

¹¹ AssistNow Offline requires external memory.



1.7 Precision Timing

1.7.1 Time mode

NEO-6T provides a special Time Mode to provide higher timing accuracy. The NEO-6T is designed for use with stationary antenna setups. The Time Mode features three different settings described in Table 3: Disabled, Survey-In and Fixed Mode. For optimal performance entering the position of the antenna (when known) is recommended as potential source of errors will be reduced.

Time Mode Settings	Description
Disabled	Standard PVT operation
Survey-In	The GPS receiver computes the average position over an extended time period until a predefined maximum standard deviation has been reached. Afterwards the receiver will be automatically set to Fixed Mode and the timing features will be activated.
Fixed Mode	In this mode, a fixed 3D position and known standard deviation is assumed and the timing features are activated. Fixed Mode can either be activated directly by feeding pre-defined position coordinates (ECEF - Earth Center Earth Fixed format) or by performing a Survey-In. In Fixed mode, the timing errors in the TIMEPULSE signal which otherwise result from positioning errors are eliminated. Single-satellite operation is supported. For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

Table 3: Time mode settings

1.7.2 Timepulse and frequency reference

NEO-6T comes with a timepulse output which can be configured from 0.25 Hz up to 10 MHz. The timepulse can either be used for time synchronization (i.e. 1 pulse per second) or as a reference frequency in the MHz range. A timepulse in the MHz range provides excellent long-term frequency accuracy and stability.

1.7.3 Time mark

NEO-6T can be used for precise time measurements with sub-microsecond resolution using the external interrupt (EXTINT0). Rising and falling edges of these signals are time-stamped to the GPS or UTC time and counted. The Time Mark functionality can be enabled with the UBXC-FGM-TM2 message

For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

1.8 Raw data

Raw data output is supported at an update rate of 5 Hz on the NEO-6T and NEO-6P. The UBXC-RXM-RAW message includes carrier phase with half-cycle ambiguity resolved, code phase and Doppler measurements, which can be used in external applications that offer precision positioning, real-time kinematics (RTK) and attitude sensing.

1.9 Automotive Dead Reckoning

Automotive Dead Reckoning (ADR) is u-blox' industry proven off-the-shelf Dead Reckoning solution for tier-one automotive customers. u-blox' ADR solution combines GPS and sensor digital data using a tightly coupled Kalman filter. This improves position accuracy during periods of no or degraded GPS signal.

The NEO-6V provides ADR functionality over its software sensor interface. A variety of sensors (such as wheel ticks and gyroscope) are supported, with the sensor data received via UBXC messages from the application processor. This allows for easy integration and a simple hardware interface, lowering costs. By using digital sensor data available on the vehicle bus, hardware costs are minimized since no extra sensors are required for Dead Reckoning functionality. ADR is designed for simple integration and easy configuration of different sensor options (e.g. with or without gyroscope) and vehicle variants, and is completely self-calibrating.



For more details contact the u-blox support representative nearest you to receive dedicated u-blox 6 Receiver Description Including Protocol Specification [3].

1.10 Precise Point Positioning

u-blox' industry proven PPP algorithm provides extremely high levels of position accuracy in static and slow moving applications, and makes the NEO-6P an ideal solution for a variety of high precision applications such as surveying, mapping, marine, agriculture or leisure activities.

Ionospheric corrections such as those received from local SBAS¹² geostationary satellites (WAAS, EGNOS, MSAS) or from GPS enable the highest positioning accuracy with the PPP algorithm. The maximum improvement of positioning accuracy is reached with PPP+SBAS and can only be expected in an environment with unobstructed sky view during a period in the order of minutes.

1.11 Oscillators

NEO-6 GPS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

1.12 Protocols and interfaces

Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

Table 4: Available protocols

All listed protocols are available on UART, USB, and DDC. For specification of the various protocols see the u-blox 6 Receiver Description including Protocol Specification [2].

1.12.1 UART

NEO-6 modules include one configurable UART interface for serial communication (for information about configuration see section 1.15).

1.12.2 USB

NEO-6 modules provide a USB version 2.0 FS (Full Speed, 12Mbit/s) interface as an alternative to the UART. The pull-up resistor on USB_DP is integrated to signal a full-speed device to the host. The VDDUSB pin supplies the USB interface. u-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

1.12.3 Serial Peripheral Interface (SPI)

The SPI interface allows for the connection of external devices with a serial interface, e.g. serial flash to save configuration and AssistNow Offline A-GPS data or to interface to a host CPU. The interface can be operated in master or slave mode. In master mode, one chip select signal is available to select external slaves. In slave mode a single chip select signal enables communication with the host.



The maximum bandwidth is 100kbit/s.

¹² Satellite Based Augmentation System



1.12.4 Display Data Channel (DDC)

The I²C compatible DDC interface can be used either to access external devices with a serial interface EEPROM or to interface with a host CPU. It is capable of master and slave operation. The DDC interface is I²C Standard Mode compliant. For timing parameters consult the I²C standard.

-  The DDC Interface supports serial communication with u-blox wireless modules. See the specification of the applicable wireless module to confirm compatibility.
-  The maximum bandwidth is 100kbit/s.

1.12.4.1 External serial EEPROM

NEO-6 modules allow an optional external serial EEPROM to be connected to the DDC interface. This can be used to store Configurations permanently.

-  For more information see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

 **Use caution when implementing since forward compatibility is not guaranteed.**

1.13 Antenna


NEO-6 modules are designed for use with passive and active¹³ antennas.

Parameter	Specification	
Antenna Type	Passive and active antenna	
Active Antenna Recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 dB
	Maximum noise figure	1.5 dB

Table 5: Antenna Specifications for all NEO-6 modules

1.14 Power Management

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption.


-  For more information about power management strategies, see the u-blox 6 Receiver Description including Protocol Specification [2].

1.14.1 Maximum Performance Mode

During a Cold start, a receiver in Maximum Performance Mode continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.

1.14.2 Eco Mode

During a Cold start, a receiver in Eco Mode works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

-  Note that even if the acquisition engine is powered off, satellites continue to be acquired.

¹³ For information on using active antennas with NEO-6 modules, see the LEA-6/NEO-6 Hardware Integration Manual [1].



1.14.3 Power Save Mode

Power Save Mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off.

 **Power Save mode is not available with NEO-6P, NEO-6T and NEO-6V.**

1.15 Configuration

1.15.1 Boot-time configuration

NEO-6 modules provide configuration pins for boot-time configuration. These become effective immediately after start-up. Once the module has started, the configuration settings can be modified with UBX configuration messages. The modified settings remain effective until power-down or reset. If these settings have been stored in battery-backup RAM, then the modified configuration will be retained, as long as the backup battery supply is not interrupted.

NEO-6 modules include both **CFG_COM0** and **CFG_COM1** pins and can be configured as seen in Table 6. Default settings in bold.

CFG_COM1	CFG_COM0	Protocol	Messages	UARTBaud rate	USB power
1	1	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	9600	BUS Powered
1	0	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	38400	Self Powered
0	1	NMEA	GSV ¹⁴ , RMC, GSA, GGA, VTG, TXT	4800	BUS Powered
0	0	UBX	NAV-SOL, NAV-STATUS, NAV-SVINFO, NAV-CLOCK, INF, MON-EXCEPT, AID-ALPSERV	57600	BUS Powered

Table 6: Supported COM settings

NEO-6 modules include a **CFG_GPS0** pin, which enables the boot-time configuration of the power mode. These settings are described in Table 7. Default settings in bold.

CFG_GPS0	Power Mode
0	Eco Mode
1	Maximum Performance Mode

Table 7: Supported CFG_GPS0 settings

 Static activation of the **CFG_COM** and **CFG_GPS** pins is not compatible with use of the SPI interface.

1.16 Design-in

In order to obtain the necessary information to conduct a proper design-in, u-blox strongly recommends consulting the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

¹⁴ Every 5th fix.



NEO-6 - Data Sheet

2 Pin Definition

2.1 Pin assignment



Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Time pulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1



NEO-6 - Data Sheet

No	Module	Name	I/O	Description
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 8: Pinout



Pins designated Reserved should not be used. For more information about Pinouts see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].



3 Electrical specifications

3.1 Absolute maximum ratings

Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDU SB	V	
DC current through any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Pfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Table 9: Absolute maximum ratings



GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. For more information see chapter 6.4.



Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes. For more information see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].



NEO-6 - Data Sheet

3.2 Operating conditions

All specifications are at an ambient temperature of 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M	2.7	3.0	3.6	V	
		NEO-6P/V/T					
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4		3.6	V	
Backup battery current	I_BCKP	All		22		μA	V_BCKP = 1.8 V, VCC = 0V
Input pin voltage range	Vin	All	0		VCC	V	
Digital IO Pin Low level input voltage	Vil	All	0		0.2*VCC	V	
Digital IO Pin High level input voltage	Vih	All	0.7*VCC		VCC	V	
Digital IO Pin Low level output voltage	Vol	All			0.4	V	Iol=4mA
Digital IO Pin High level output voltage	Voh	All	VCC -0.4			V	Ioh=4mA
USB_DM, USB_DP	VinU	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All		VCC-0.1		V	
VCC_RF output current	ICC_RF	All			50	mA	
Antenna gain	Gant	All			50	dB	
Receiver Chain Noise Figure	NFtot	All		3.0		dB	
Operating temperature	Topr	All	-40		85	°C	

Table 10: Operating conditions

Operation beyond the specified operating conditions can affect device reliability.

3.3 Indicative power requirements

Table 11 lists examples of the total system supply current for a possible application.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current ¹⁵	Iccp	All			67	mA	VCC = 3.6 V ¹⁶ / 1.95 V ¹⁷
	Icc Acquisition	All		47 ¹⁹		mA	
	Icc Tracking	NEO-6G/Q/T		40 ²⁰		mA	
	(Max Performance mode)	NEO-6M/P/V		39 ²⁰		mA	
Average supply current ¹⁸	Icc Tracking	NEO-6G/Q/T		38 ²⁰		mA	VCC = 3.0 V ¹⁶ / 1.8 V ¹⁷
	(Eco mode)	NEO-6M/P/V		37 ²⁰		mA	
	Icc Tracking	NEO-6G/Q		12 ²⁰		mA	
	(Power Save mode / 1 Hz)	NEO-6M		11 ²⁰		mA	

Table 11: Indicative power requirements

Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

¹⁵ Use this figure to dimension maximum current capability of power supply. Measurement of this parameter with 1 Hz bandwidth.

¹⁶ NEO-6Q, NEO-6M, NEO-6P, NEO-6V, NEO-6T

¹⁷ NEO-6G

¹⁸ Use this figure to determine required battery capacity.

¹⁹ >8 SVs in view, CNo >40 dBHz, current average of 30 sec after cold start.

²⁰ With strong signals, all orbits available. For Cold Starts typical 12 min after first fix. For Hot Starts typical 15 s after first fix.



3.4 SPI timing diagrams

In order to avoid a faulty usage of the SPI, the user needs to comply with certain timing conditions. The following signals need to be considered for timing constraints:

Symbol	Description
SS_N	Slave Select signal
SCK	Slave Clock signal

Table 12: Symbol description

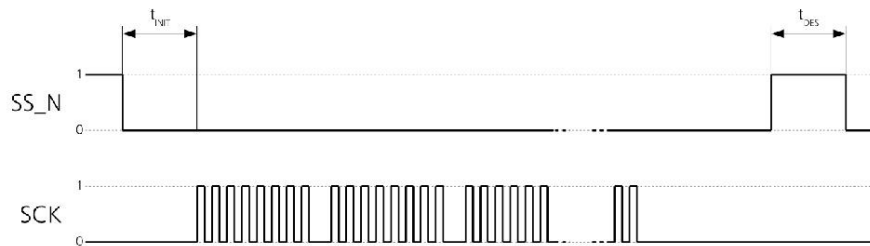


Figure 3: SPI timing diagram

3.4.1 Timing recommendations

Parameter	Description	Recommendation
t_{INIT}	Initialization Time	500 μ s
t_{DES}	Deselect Time	1 ms
Bitrate		100 kbit/s

Table 13: SPI timing recommendations



The values in the above table result from the requirement of an error-free transmission. By allowing just a few errors, the byte rate could be increased considerably. These timings – and therefore the byte rate – could also be improved by disabling other interfaces, e.g. the UART.



The maximum bandwidth is 100 kbit/s²¹.

²¹ This is a theoretical maximum, the protocol overhead is not considered.



4 Mechanical specifications

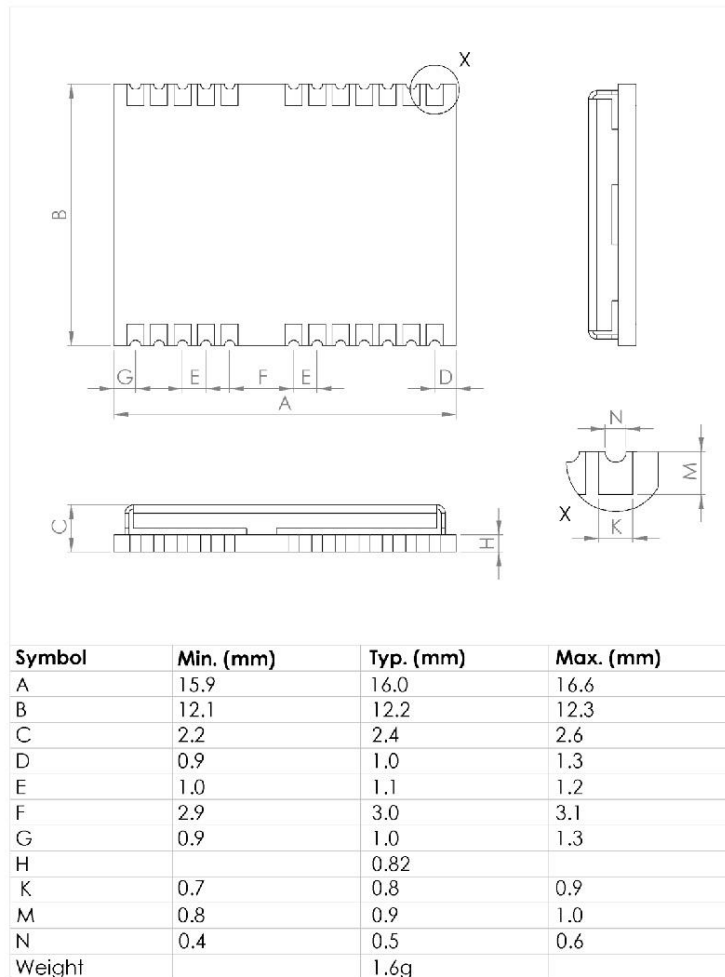


Figure 4: Dimensions



For information regarding the Paste Mask and Footprint see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].



5 Qualification and certification

5.1 Reliability tests



All NEO-6 modules are based on AEC-Q100 qualified GPS chips.

Tests for product family qualifications according to ISO 16750 "Road vehicles - Environmental conditions and testing for electrical and electronic equipment", and appropriate standards.

5.2 Approvals



Products marked with this lead-free symbol on the product label comply with the "Directive 2002/95/EC of the European Parliament and the Council on the Restriction of Use of certain Hazardous Substances in Electrical and Electronic Equipment" (RoHS).

All u-blox 6 GPS modules are RoHS compliant.



6 Product handling & soldering

6.1 Packaging

NEO-6 modules are delivered as hermetically sealed, reeled tapes in order to enable efficient production, production lot set-up and tear-down. For more information about packaging, see the u-blox Package Information Guide [4].



Figure 5: Reeled u-blox 6 modules

6.1.1 Reels

NEO-6 GPS modules are deliverable in quantities of 250pcs on a reel. NEO-6 modules are delivered using reel Type B as described in the u-blox Package Information Guide [4].

Parameter	Specification
Reel Type	B
Delivery Quantity	250

Table 14: Reel information for NEO-6 modules



NEO-6 - Data Sheet

6.1.1 Tapes

Figure 6 shows the position and orientation of NEO-6 modules as they are delivered on tape. The dimensions of the tapes are specified in Figure 7.

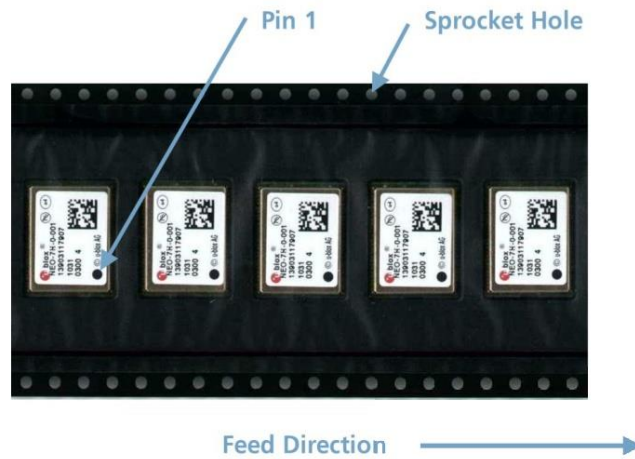


Figure 6: Orientation for NEO-6 modules on tape

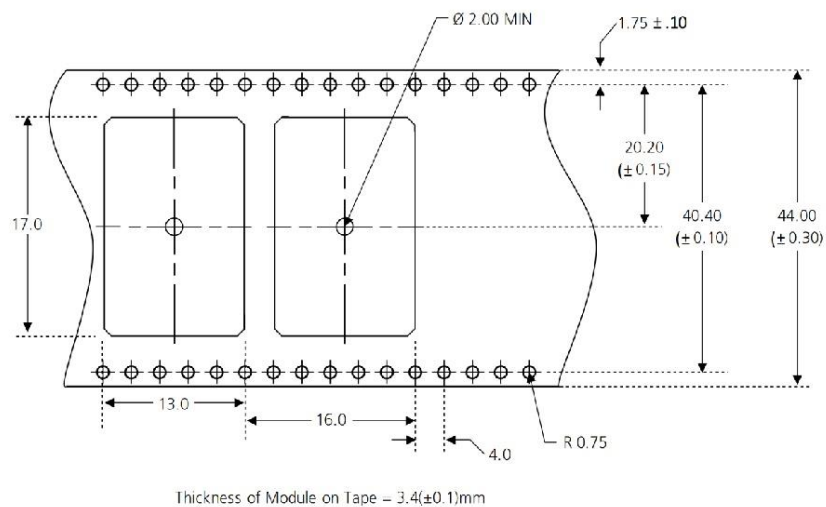


Figure 7: NEO tape dimensions (mm)



6.2 Moisture Sensitivity Levels



NEO-6 modules are Moisture Sensitive Devices (MSD) in accordance to the IPC/JEDEC specification.

NEO-6 modules are rated at MSL level 4. For more information regarding moisture sensitivity levels, labeling, storage and drying see the u-blox Package Information Guide [4].



For MSL standard see IPC/JEDEC J-STD-020, which can be downloaded from www.jedec.org.

6.3 Reflow soldering

Reflow profiles are to be selected according to u-blox recommendations (see LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1]).

6.4 ESD handling precautions



NEO-6 modules contain highly sensitive electronic circuitry and are Electrostatic Sensitive Devices (ESD). Observe precautions for handling! Failure to observe these precautions can result in severe damage to the GPS receiver!



GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. Particular care must be exercised when handling patch antennas, due to the risk of electrostatic charges. In addition to standard ESD safety practices, the following measures should be taken into account whenever handling the receiver:

- Unless there is a galvanic coupling between the local GND (i.e. the work table) and the PCB GND, then the first point of contact when handling the PCB must always be between the local GND and PCB GND.
- Before mounting an antenna patch, connect ground of the device
- When handling the RF pin, do not come into contact with any charged capacitors and be careful when contacting materials that can develop charges (e.g. patch antenna ~10pF, coax cable ~50-80pF/m, soldering iron, ...)
- To prevent electrostatic discharge through the RF input, do not touch any exposed antenna area. If there is any risk that such exposed antenna area is touched in non ESD protected work area, implement proper ESD protection measures in the design.
- When soldering RF connectors and patch antennas to the receiver's RF pin, make sure to use an ESD safe soldering iron (tip).





7 Default settings

Interface	Settings
Serial Port 1 Output	9600 Baud, 8 bits, no parity bit, 1 stop bit Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT (In addition to the 6 standard NMEA messages the NEO-6T includes ZDA).
USB Output	Configured to transmit both NMEA and UBX protocols, but only following NMEA and no UBX messages have been activated at start-up: GGA, GLL, GSA, GSV, RMC, VTG, TXT (In addition to the 6 standard NMEA messages the NEO-6T includes ZDA). USB Power Mode: Bus-Powered
Serial Port 1 Input	9600 Baud, 8 bits, no parity bit, 1 stop bit Automatically accepts following protocols without need of explicit configuration: UBX, NMEA The GPS receiver supports interleaved UBX and NMEA messages.
USB Input	Automatically accepts following protocols without need of explicit configuration: UBX, NMEA The GPS receiver supports interleaved UBX and NMEA messages. USB Power Mode: Bus-Powered
TIMEPULSE (1Hz Nav)	1 pulse per second, synchronized at rising edge, pulse length 100ms
Power Mode	Maximum Performance mode
AssistNow Autonomous	Disabled.

Table 15: Default settings

Refer to the u-blox 6 Receiver Description including Protocol Specification [2] for information about further settings.



8 Labeling and ordering information

8.1 Product labeling

The labeling of u-blox 6 GPS modules includes important product information. The location of the product type number is shown in Figure 8.

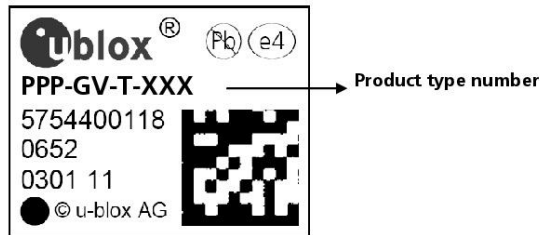


Figure 8: Location of product type number on u-blox 6 module label

8.2 Explanation of codes

3 different product code formats are used. The **Product Name** is used in documentation such as this data sheet and identifies all u-blox 6 products, independent of packaging and quality grade. The **Ordering Code** includes options and quality, while the **Type Number** includes the hardware and firmware versions. Table 16 below details these 3 different formats:

Format	Structure
Product Name	PPP-GV
Ordering Code	PPP-GV-T
Type Number	PPP-GV-T-XXX

Table 16: Product Code Formats

The parts of the product code are explained in Table 17.

Code	Meaning	Example
PPP	Product Family	NEO
G	Product Generation	6 = u-blox6
V	Variant	T = Timing, R = DR, etc.
T	Option / Quality Grade	Describes standardized functional element or quality grade such as Flash size, automotive grade etc.
XXX	Product Detail	Describes product details or options such as hard- and software revision, cable length, etc.

Table 17: part identification code



NEO-6 - Data Sheet

8.3 Ordering information

Ordering No.	Product
NEO-6G-0	u-blox 6 GPS Module, 1.8V, TCXO, 12x16mm, 250 pcs/reel
NEO-6M-0	u-blox 6 GPS Module, 12x16mm, 250 pcs/reel
NEO-6Q-0	u-blox 6 GPS Module, TCXO, 12x16mm, 250 pcs/reel
NEO-6P-0	u-blox 6 GPS Module, PPP, 12x16mm, 250 pcs/reel
NEO-6V-0	u-blox 6 GPS Module, Dead Reckoning SW sensor, 12x16mm, 250 pcs/reel
NEO-6T-0	u-blox 6 GPS Module, Precision Timing, TCXO, 12x16mm, 250 pcs/reel

Table 18: Product Ordering Codes



Product changes affecting form, fit or function are documented by u-blox. For a list of Product Change Notifications (PCNs) see our website at: <http://www.u-blox.com/en/notifications.html>

Related documents

- [1] LEA-6/NEO-6/MAX-6 Hardware Integration Manual, Docu. GPS.G6-HW-09007
- [2] u-blox 6 Receiver Description Including Protocol Specification (Public version), Docu. No. GPS.G6-SW-10018
- [3] u-blox 6 Receiver Description Including Protocol Specification (Confidential version), Docu. No. GPS.G6-SW-10019
- [4] u-blox Package Information Guide, Docu. No GPS-X-11004



For regular updates to u-blox documentation and to receive product change notifications please register on our homepage.

Revision history

Revision	Date	Name	Status / Comments
	31/08/2009	tgri	Initial Version
1	21/09/2009	tgri	update of section 1.3 GPS performance, section 1.4 block diagram, section 3.2 peak supply current
A	25/02/2010	tgri	Change of status to Advance Information. Addition of NEO-6G. Update of section 1.8.2, removed reference to Vddio – added USB driver certification. Update of section 3.2 table 11: average supply current, Added section 3.3-3.4: SPI & DDC timing, section 5.1: addition of table 12.
B	24/06/2010	dhur	Change of status to Preliminary. Update of section 1.2, 1.8.4, 1.10.4, 3.1, 3.2 and chapter 2 and 4. General clean-up and consistency check.
B1	11/08/2010	dhur	Replaced graphic in figure 2.
C	18/07/2011	dhur	Added chapter 1.6, update to FW7.03.
D	19/10/2011	dhur	Added NEO-6P and NEO-6V. Added chapter 1.7 and 1.8. Revised Chapter 6.
E	05/12/2011	dhur	Added NEO-6T. Added chapter 1.7 and 1.8. Added Accuracy for Timepulse signal in Table 2. Corrected Maximum Input power at RF_IN for NEO-6P in Table 9.



Contact

For complete contact information visit us at www.u-blox.com

Headquarters

u-blox AG

Zuercherstrasse 68
CH-8800 Thalwil
Switzerland

Phone: +41 44 722 74 44
Fax: +41 44 722 74 47
E-mail: info@u-blox.com

Offices

North, Central and South America

u-blox America, Inc.

Phone: +1 (703) 483 3180
E-mail: info_us@u-blox.com

Regional Office West Coast:

Phone: +1 (703) 483 3184
E-mail: info_us@u-blox.com

Technical Support:

Phone: +1 (703) 483 3185
E-mail: support_us@u-blox.com

Europe, Middle East, Africa

u-blox AG

Phone: +41 44 722 74 44
E-mail: info@u-blox.com

Technical Support:

Phone: +41 44 722 74 44
E-mail: info@u-blox.com

Asia, Australia, Pacific

u-blox Singapore Pte. Ltd.

Phone: +65 6734 3811
E-mail: info_ap@u-blox.com
Support: support_ap@u-blox.com

Regional Office China:

Phone: +86 10 68 133 545
E-mail: info_cn@u-blox.com
Support: support_cn@u-blox.com

Regional Office Japan:

Phone: +81 3 5775 3850
E-mail: info_jp@u-blox.com
Support: support_jp@u-blox.com

Regional Office Korea:

Phone: +82 2 542 0861
E-mail: info_kr@u-blox.com
Support: support_kr@u-blox.com

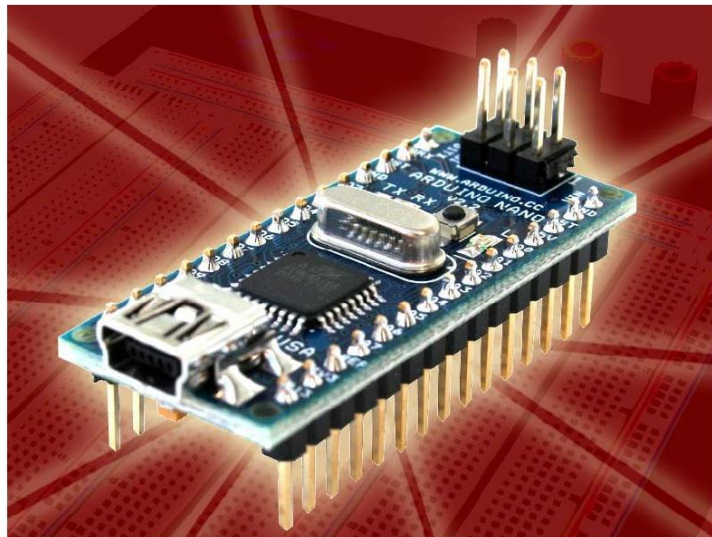
Regional Office Taiwan:

Phone: +886 2 2657 1090
E-mail: info_tw@u-blox.com
Support: support_tw@u-blox.com

6. Arduino nano – DATASHEET [18]

Arduino Nano (V2.3)

User Manual



Released under the Creative Commons Attribution Share-Alike 2.5 License

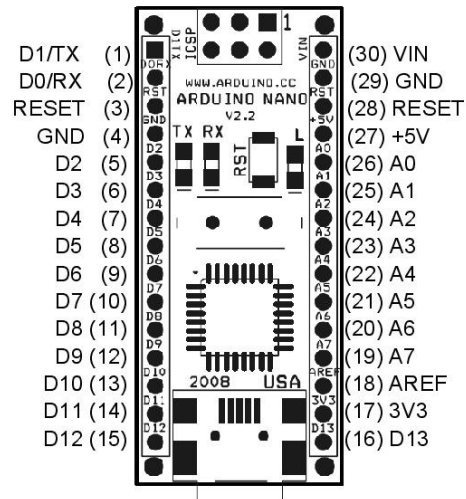
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

www.arduino.cc

Rev. 2.3

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino Nano Bill of Material

Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805	C0805C180J5GACTU	Kemet	80-C0805C180J5G	Mouser
4	1	D1	Diode, Schottky 0.5A 20V	MBR0520LT1G	ONsemi	863-MBR0520LT1G	Mouser
5	1	J1,J2	Headers, 36PS 1 Row	68000-136HLF	FCI	649-68000-136HLF	Mouser
6	1	J4	Connector, Mini-B Recept Rt. Angle	67503-1020	Molex	538-67503-1020	Mouser
7	1	J5	Headers, 72PS 2 Rows	67996-272HLF	FCI	649-67996-272HLF	Mouser
8	1	LD1	LED, Super Bright RED 100mcd 640nm 120degree 0805	APT2012SRCPRV	Kingbright	604-APT2012SRCPRV	Mouser
9	1	LD2	LED, Super Bright GREEN 50mcd 570nm 110degree 0805	APHCM2012CGCK-F01	Kingbright	604-APHCM2012CGCK	Mouser
10	1	LD3	LED, Super Bright ORANGE 160mcd 601nm 110degree 0805	APHCM2012SECK-F01	Kingbright	04-APHCM2012SECK	Mouser
11	1	LD4	LED, Super Bright BLUE 80mcd 470nm 110degree 0805	LTST-C170T8T	Lite-On Inc	160-1579-1-ND	Digikey
12	1	R1	Resistor Pack, 1K +/-5% 62.5mW 4RES SMD	YC164-JR-071KL	Yageo	YC164J-1.0KCT-ND	Digikey
13	1	R2	Resistor Pack, 680 +/-5% 62.5mW 4RES SMD	YC164-JR-07680RL	Yageo	YC164J-680CT-ND	Digikey
14	1	SW1	Switch, Momentary Tact SPST 150gf 3.0x2.5mm	B3U-1000P	Omron	SW1020CT-ND	Digikey
15	1	U1	IC, Microcontroller RISC 16kB Flash, 0.5kB EEPROM, 23 I/O Pins	ATmega168-20AU	Atmel	556-ATMEGA168-20AU	Mouser
16	1	U2	IC, USB to SERIAL UART 28 Pins SSOP	FT232RL	FTDI	895-FT232RL	Mouser
17	1	U3	IC, Voltage regulator 5V, 500mA SOT-223	UA78M05CDCYRG3	TI	595-UA78M05CDCYRG3	Mouser
18	1	Y1	Crystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abracon	815-ABL-16-B2	Mouser

